UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Paulo Alceu d' Almeida Rezende

BROAD-RS: arquitetura para recomendação de objetos de aprendizagem sensível ao contexto usando agentes e ontologia

Juiz de Fora
2014

UNIVERSIDADE FEDERAL DE JUIZ DE FORA

INSTITUTO DE CIÊNCIAS EXATAS

PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Paulo Alceu d' Almeida Rezende

BROAD-RS: arquitetura para recomendação de objetos de aprendizagem sensível ao contexto usando agentes e ontologia

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientadora: Fernanda Claudia Alves Campos, D. Sc.

Juiz de Fora

2014

DEDICO ESTE TRABALHO AO MEU PAI,
ALCEU CASALI DE REZENDE, MEU EXEMPLO.

RESUMO

Objetos de aprendizagem são quaisquer recursos que possam ajudar no processo de ensino e aprendizagem sendo reutilizados em diversos contextos. Essa reutilização de objetos de aprendizagem mostra seu potencial para acelerar a preparação e a composição de cursos *on line*. Um sistema de recomendação na área educacional tem por objetivo identificar o perfil do aluno para que seja possível sugerir objetos de aprendizagem adequados às suas preferências. Entretanto, ao considerar o reuso de conteúdos, também se observa a necessidade de adaptação dos mesmos. Aplicações cientes de contexto são aplicações que são capazes de modificar seu comportamento baseado nas informações dos usuários. Uma motivação de usar um sistema sensível ao contexto é garantir a mobilidade transparente e fazer com que aplicações estejam de acordo com os elementos do ambiente. Este trabalho apresenta a arquitetura BROAD-RS (*BROAD Recommendation System*) capaz de realizar a recomendação de objetos de aprendizagem sensível ao contexto, baseado em uma ontologia para modelagem do perfil e contexto do aluno em um ambiente *e-learning* e implementado em um sistema multi-agentes.

Palavras-chave: Objetos de Aprendizagem, Recomendação de Objetos de Aprendizagem, Contexto, Agentes, Ontologia.

ABSTRACT

Learning objects are any resources that might help in the teaching and learning process and to be reused in different contexts. The reuse of learning objects shows its potential to accelerate the preparation and composition of on line courses. A recommendation system in education aims to identify student profile so that it can suggest learning objects suitable to his/her preference. However, when considering content reuse, it is also necessary to adapt them. Context-aware applications are able to modify their behavior based on the users´ informations. One motivation of using a context-sensitive system is to assure seamless mobility and make applications comply with environment elements. This paper presents BROAD-RS (BROAD Recommendation System) architecture capable of performing context sensitive recommendation of learning objects based on an ontology for modeling the profile and context of the student in an e-learning environment implemented in a multi-agent system.

Keywords: *Learning Objects, Learning Objects Recommendation, Context, Agents, Ontology.*

AGRADECIMENTOS

Primeiramente, agradeço a Deus, por ter me proporcionado saúde e entusiasmo. Agradeço a minha amada mãe, Terezinha Marlene d´ Almeida Rezende, pelo apoio total e irrestrito. Agradeço a minha amada esposa, Raquel Bittar Ramos P. Rezende, pela força que me deu em todos os momentos.

Meus agradecimentos a todos os professores do Programa de Pós-Graduação em Ciência de Computação, em especial a minha orientadora Fernanda Campos, pela paciência e dedicação.

A todos os alunos que estudaram comigo no Mestrado em Ciência da Computação da Universidade Federal de Juiz de Fora.

A todos meus colegas de trabalho e grandes amigos do CEAD/UFJF que sempre me incentivaram nesta jornada.

A todos os amigos que me incentivaram e a todos aqueles que de alguma forma contribuíram para a realização deste trabalho.

Por fim, meus agradecimentos a todos aqueles que, direta ou indiretamente, contribuíram para o desenvolvimento desse trabalho.

SUMÁRIO

1 Introdução 1
1.1 Metodologia 3
1.2 Hipóteses 3
1.3 Objetivos 4
1.4 Organização 4
2 Objetos de Aprendizagem 5
2.1 Aspectos Pedagógicos 9
2.2 Padrões de Objetos de Aprendizagem 10
2.2.1 Padrão SCORM 11
2.2.2 Padrão LOM-IEEE 12
2.2.3 Padrão OBAA 14
2.2.4 Padrão BROAD 16
2.3 Considerações Finais do Capítulo 18
3 Recomendação, Ontologias e Modelos de Alunos 19
3.1 Sistemas de Recomendação 19
3.1.1 Filtragem Colaborativa 21
3.1.2 Filtragem Baseada em Conteúdo 22
3.1.3 Filtragem Híbrida 22
3.1.4 Filtragem Baseada em Mineração de Dados 23
3.1.5 Raciocínio Baseado em Casos 23
3.2 Contexto 24
3.2.1 Sensibilidade ao Contexto 24
3.2.2 Ambientes Sensíveis ao Contexto 28
3.3 Rede de Ontologias 29
3.4 Padrões e Modelos de Dados de Alunos 31
3.4.1 EduPerson 32
3.4.2 Universal Learning Format 33
3.4.3 IEEE PAPI *(Public and Private Information Learner)* 34
3.4.4 *IMS Learner Information Package* 36
3.4.5 Análise compartiva dos padrões e modelos 40
3.5 Considerações Finais do Capítulo 43
4 Sistemas Baseados em Agentes 45
4.1 Agentes 45
4.2 Modelos para a Contrução de Agentes e seus Comportamentos 49
4.2.1 Modelo BDI (*Beliefs, Desires and Intentions*) 49
4.2.2 Modelo 5C 51

4.3 Desenvolvimento de Sistemas Multiagentes 52
4.3.1 Metodologia INGENIAS 53
4.4 *Framework* para Agentes 59
4.4.1 Jadex 59
4.4.2 JADE 62
4.5 Considerações Finais do Capítulo 65
5 Trabalhos Relacionados 67
5.1 Modelos e Propostas de Recomendação 67
5.2 Propostas de Modelos de Dados de Alunos 69
5.2.1 Proposta de Chen e Mizoguchi 69
5.2.2 Proposta de Brusilovsky e Peylo 70
5.2.3 Proposta de Niederée 71
5.2.4 Proposta de Dolog e Nejdl 72
5.2.5 Proposta de Sudhana et al. 72
5.2.6 Proposta de Alharbi et al. 74
5.2.7 Proposta de Klašnja-Milicevic et al. 75
5.2.8 Proposta de Wang e Wu 76
5.2.9 Proposta de Fritzen et al. 77
5.2.10 Proposta de Dwivedi e Bharadwaj 78
5.2.11 Proposta de Gasparini et al. 80
5.2.12 Proposta de Verbert et al. 80
5.3 Considerações Finais do Capítulo 83
6 BROAD-RS: Sistema para Recomendação de OA Sensível ao Contexto 84
6.1 Projeto BROAD 84
6.2 Arquitetura BROAD-RS 87
6.3 Plano Didático do Professor 88
6.4 Agentes da Arquitetura 90
6.4.1 Projeto e especificação dos agentes 92
6.5 Ontologia PERSONNA 102
6.5.1 Situações de Aprendizagem 108
6.5.2 Representação do Contexto na Ontologia 109
6.5.3 Formalização das Regras de Inferência 110
6.6 Protótipo 112
6.7 Cenários de Aprendizagem 113
6.7.1 Cenário 1 –Professor/Plano Didático e Aluno/Contexto 114
6.8 Análise do Cenário Educacional Apresentado 122
6.9 Análises Relativas aos Agentes 124
6.9.1 Cenário 2 – Professor acessando sistema 124
6.9.2 Cenário 3 – Professor fazendo alterações no sistema (Plano Didático) 124
6.9.3 Cenário 4 – Aluno fazendo alterações no sistema (Perfil e Contexto) 125

6.9.4 Cenário 5 – Aluno acessando o sistema 126
6.10 Considerações Finais do Capítulo 126
7 Considerações Finais 128

LISTA DE TABELAS

Tabela 1: Conjunto mínimo de atributos necessários LOM (IEEE, 2012) e (CAMPOS et al. 2011b) 12
Tabela 2: Metadados BROAD (CAMPOS et al. 2011b) 16
Tabela 3: Metadados Educacionais BROAD 17
Tabela 4: Atributos do EduPerson (EDUPERSON, 2005) 32
Tabela 5: Atributos do ULF (SABA, 2000) 33
Tabela 6: Atributos do PAPI (PAPI, 2001) 35
Tabela 7: Atributos do LIP (IMS LIP, 2007) 38
Tabela 8: Relação entre as categorias dos modelos 40
Tabela 9: Comparativo entre os modelos 41
Tabela 10: Síntese dos Trabalhos Relacionados 68
Tabela 11: Elementos utilizados nos trabalhos referenciados 81
Tabela 12: Metadados do Plano Didático da Arquitetura BROAD-RS 89
Tabela 13: Categorias e Propriedades da Ontologia PERSONNA e seus detalhes 103
Tabela 14: Parte da Tabela de Relações de Porcentagem para contrução de regras 112

LISTA DE FIGURAS

Figura 1: Anatomia de um Objeto de Aprendizagem (JOHNSTON, 2003) 5
Figura 2: Relação dos objetos de aprendizagem com as três grandes áreas envolvidas em sistemas de *e-learning*..... 6
Figura 3: Estrutura de um curso (MORALES; AGÜERA, 2012)..... 8
Figura 4: Taxonomia de Objetos de Aprendizagem (HODGINS, 2001)..... 8
Figura 5: SCORM como conjunto de especificações (DUTRA; TAROUCO, 2006) 12
Figura 6: Elementos e estrutura do esquema conceitual de dados do LOM extraído de (http://www.imsglobal.org/metadata/mdv1p3pd/imsmd_bestv1p3pd.html) 13
Figura 7: OAs nas plataformas Web, TV Digital e dispositivos móveis (VICARI et al. 2009) 14
Figura 8: Categoria Educational (VICARI et al. 2009) 15
Figura 9: Categorias *Acessibility* (VICARI et al. 2009) 15
Figura 10: Metadados essenciais BROAD no segundo nível de detalhamento 17
Figura 11. Filtragem híbrida (CAZELLA et al. 2010)..... 23
Figura 12: Sugestão de classificação dos tipos de contexto (FLEISCHMANN, 2012) 26
Figura 13: Categorias PAPI (PAPI, 2001) 34
Figura 14: Categorias do LIP 37
Figura 15: Interação do agente e o ambiente (RUSSEL; NORVIG, 1995) 47
Figura 16: Diagrama de uma arquitetura *belief-desire-intention* genérica (WOOLDRIDGE, 1999) 50
Figura 17: As cinco visões para SMA definidas com INGENIAS..... 54
Figura 18: Ambiente IDK - INGENIAS Development Kit..... 58
Figura 19: Elementos relevantes da notação INGENIAS 59
Figura 20: *INGENIAS Development Kit (IDK) (http://grasia.fdi.ucm.es)*..... 59
Figura 21: Arquitetura abstrata Jadex (POKAHR; BRAUBACH, 2007)..... 60
Figura 22: Plataformas e containeres no JADE 4.1 63
Figura 23: Arquitetura da plataforma JADE (BELLIFEMINE, 2003)..... 64
Figura 24: Ontologia do modelo de aluno proposta por Chen e Mizoguchi (1999)..... 70
Figura 25: Ontologia proposta por Brusilovsky e Peylo (2003)..... 71
Figura 26: Modelo UUCM (NIEDERÉE et al. 2005)..... 71
Figura 27: Ontologia para perfil de aluno proposta por Dolog e Nejdl (2007)..... 72

Figura 28: Ontologia parcial representando as três principais categorias de informações contextuais (SUDHANA et al. 2013) 73
Figura 29: Visão geral da proposta arquitetural (SUDHANA et al. 2013) 74
Figura 30: Arquitetura proposta por Alharbi et al. (2012) 75
Figura 31: O componente de recomendação do sistemas Protus (KLAŠNJA-MILICEVIC et al. 2011) 76
Figura 32: Sistema ubíquo adaptativo (WANG; WU, 2011) 77
Figura 33: Visão geral da arquitetura de Fritzen et al. (2012) 78
Figura 34: Associação por transitividadem (DWIVEDI; BHARADWAJ, 2012) 79
Figura 35: Arquitetura estendida do Ambiente AdaptWeb® (GASPARINI et al. 2011) 80
Figura 36: *Framework* de contexto (VERBERT et al. 2012) 81
Figura 37: Ontologia BROAD (NERY et al. 2011) 85
Figura 38: Classificação das principais ontologias da rede BROADNet (CAMPOS et al. 2012b) 86
Figura 39: Arquitetura Proposta para o Projeto BROAD-PL (CAMPOS et al. 2012b) 86
Figura 40: Arquitetura BROAD-RS 87
Figura 41: Diagrama de Casos de Uso: Geral 93
Figura 42: Diagrama de Casos de Uso: Específico 1 93
Figura 43: Diagrama de Casos de Uso: Específico 2 93
Figura 44: Modelo de Organização – Geral 94
Figura 45: Modelo de Organização – Específico 1 94
Figura 46: Modelo de Agentes – *Student Agent* 95
Figura 47: Modelo de Agentes – *Planner Agent* 95
Figura 48: Modelo de Agentes – *Teacher Agent* 95
Figura 49: Modelo de Agentes - *AdminAgent* 96
Figura 50: Modelo de Tarefas e Objetivos - Nível Macro 96
Figura 51: Modelo de Tarefas e Objetivos - Gerenciamento 97
Figura 52: Modelo de Tarefas e Objetivos – Recomendação 97
Figura 53: Modelo de Interação - Interação para o registro nas "páginas amarelas" 98
Figura 54: Modelo de Interação - Interação para o registro das "páginas brancas" 98
Figura 55: Modelo de Interação - Troca de mensagens entre o *DFAgent* e *StudentAgent* 99
Figura 56: Modelo de Interação - Troca de mensagens entre o *AMSAgent* e o *StudentAgent* . 99
Figura 57: Modelo de Interação - Interação para a busca dos objetos de aprendizagem 1 100
Figura 58: Modelo de Interação - Interação para a busca dos objetos de aprendizagem 2 100

Figura 59: Modelo de Interação - Troca de mensagens entre *StudentAgent* e *AdminAgent* .. 100
Figura 60: Modelo de Interação - Troca de mensagens entre *AdminAgent* e *PlannerAgent* . 101
Figura 61: Modelo de Interação - Interação entre o *StudentAgent* e *PlannerAgent* para a busca dos OA 101
Figura 62: Modelo de Interação - Troca de mensagens entre o *StudentAgent* e *PlannerAgent* 101
Figura 63: Modelo de Ambiente - Geral 102
Figura 64: Visão geral da ontologia PERSONNA apresentando algumas de suas classes.... 103
Figura 65: Ontologia PERSONNA aberta na Ferramenta Protegé..... 107
Figura 66: Visão Parcial das Principais Classes da Ontologia PERSONNA..... 107
Figura 67 Visão Parcial das Principais Classes da Ontologia de Learning Objects..... 108
Figura 68. Generalização das situações de aprendizagem 109
Figura 69: Amostra das Regras SWRL da Ontologia PERSONNA..... 111
Figura 70: Tela do acompanhamento dos agentes no desenvolvimento do sistema..... 113
Figura 71: O RMA (*Remote Manager Agents*) 113
Figura 72: Tela de Login do BROAD-RS..... 114
Figura 73: Tela de entrada do sistema com acesso tipo Professor 115
Figura 74: Tela de Alterar Senha..... 115
Figura 75: Cadastro de Objetos de Aprendizagem (Parte 1 de 3) 116
Figura 76: Cadastro de Objetos de Aprendizagem (Parte 2 de 3) 117
Figura 77: Cadastro de Objetos de Aprendizagem (Parte 3 de 3) 118
Figura 78: Tela de Definição do Plano Didático do Professor..... 119
Figura 79: Tela de Entrada do Sistema com Acesso tipo Aluno..... 120
Figura 80: Tela de perfil e contexto do aluno (Perfil 1)..... 120
Figura 81: Tela de recomendação de OA do BROAD-RS (Primeira Recomendação) 121
Figura 82: Tela de perfil e contexto do aluno (Perfil 2)..... 122
Figura 83: Tela de recomendação de OA do protótipo BROAD-RS (Segunda Recomendação) 122
Figura 84: Cenário de Aprendizagem 2 - Professor acessando sistema 124
Figura 85: Cenário de Aprendizagem 3 - Professor criando seu plano didático..... 125
Figura 86: Cenário de Aprendizagem 4 - Aluno fazendo alterações no sistema (Perfil e Contexto)..... 125
Figura 87: Cenário de Aprendizagem 5 - Aluno acessando o sistema 126

LISTA DE ABREVIAÇÕES

ADL	*Advanced Distributed Learning Initiative*
AICC	*Aviation IndustryCBT (Computer Based Training Committee)*
API	*Application Program Interface*
ARIADNE	*Alliance of Remote Instrucional Authoring and Distribution Networks for Europe*
AUML	*Agents Unified Modeling Language*
AVA	Ambiente Virtual de Aprendizagem
BROAD	Busca e Recuperação de Objetos de Aprendizagem a Distância
CEAD	Centro de Educação a Distância
CRC	Classes, Responsabilidades e Colaborações
DBMS	*DataBase Management System*
DC	*Dublin Core*
DCC	Departamento de Ciência da Computação
DCMI	*Dublin Core Metadata Initiative*
DNVP	*Dotted Name-Value Pair*
DTD	*Document Type Definition*
EAD	Educação à Distância
FAURGS	Fundação de Apoio da Universidade Federal do Rio Grande do Sul
FBC	Filtragem Baseada em Conteúdo
FC	Filtragem Colaborativa
FHC	Filtragem Híbrida Conteúdo
GOPRR	*Graph, Object, Property, Relationship, and Role*
IEEE	*Institute of Electrical and Electronic Engineers*
IIOP	*Internet Inter-Orb Protocol*
IMS	*Instructional Management System*
INGENIAS	*Engineering for Software Agents*
JADE	*Java Agent Developing Framework*
JDBC	*Java DataBase Connection*
JSP	*Java Server Pages*
KQML	*Knowledge Query Manipulation Language*
LIP	*Learner Information Package*

LMS	*Learning Management System*
LOM	*Learning Object Metadata*
LO	*Learning Object*
LPS	Linha de Produtos de Software
LTSC	*Learning Technology Standards Committee*
MaSE	*Multiagent Systems Engineering*
MILOS	*Multiagent Infrastructure for Learning Object Support*
NLII	*US National Learning Infrastructure Initiative*
OA	Objetos de Aprendizagem
OBAA	Objetos de Aprendizagem Baseados em Agentes
OWL DL	*Web Ontology Language - Description Logic*
PDF	*Portable Document Format*
PLEs	Recomendação em ambientes pessoais de aprendizagem
QCL	Qualificações, Certificados e Licenças
RDF	*Resource Description Framework*
RMI	*Remote Method Invocation*
SCORM	*Sharable Content Object Reference Model*
SMA	Sistema Multi-agente
UFJF	Universidade Federal de Juiz de Fora
UMBRELO	*Ubiquitous Metadata Based Reusable Learning Objects*
URI	*Uniform Resource Identiler*
URL	*Uniform Resource Locator*
UUCM	*Unified User Context Model*
W3C	*World Wide Web Consortium*
XML	*Extensible Markup Language*
XSLT	*Extensible Stylesheet Language Transformation*

1 INTRODUÇÃO

Um grande desafio é poder proporcionar educação de mais qualidade a uma quantidade cada vez maior de pessoas. Particularmente na EaD (Educação a Distância), a tecnologia de informação tem tido um papel fundamental. Cada vez mais as instituições de ensino estão oferecendo inúmeras oportunidades com um mínimo de presença física dos alunos e professores. Mais e mais, as interações entre as pessoas e os objetos do conhecimento estão sendo mediadas pela tecnologia de informação.

Historicamente existem áreas do conhecimento nas quais os alunos apresentam dificuldades no processo de aprendizagem e nesse contexto, constantemente ocorrem desníveis na formação destes alunos. Com o *e-learning* surge a possibilidade dos alunos determinarem o conteúdo, a escolha dos processos de avaliação, o ritmo de aprendizado e a utilização de um método adequado, o que pode auxiliar a resolver estas dificuldades de aprendizado. Notadamente, no ensino superior, o *e-learning* tem tido um papel importante, considerando as universidades virtuais, cursos *online* e qualificação de trabalhadores (MAGANO, 2008).

Assim, com o advento da web, benefícios para a educação surgiram como consequência de sua natureza distribuída e sua facilidade para armazenar e entregar informações surgindo assim, uma nova geração de sistemas de ensino. No contexto do *e-learning*, o computador contribuiu para formar a base tecnológica para uma nova geração de sistemas de ensino, que têm sofrido uma grande sofisticação no uso das tecnologias da informação e de telecomunicações. Existe a necessidade de avanço nas atividades de ensino e aprendizagem e que os OA (Objetos de Aprendizagem) possam ser acessíveis pela Web e serem adaptáveis aos diversos dispositivos.

Um dos desafios da educação a distância é a adequação do conteúdo ao estudante de acordo com as suas características. Esta é uma tarefa difícil de ser realizada, pois cada professor deve conhecer as necessidades e competências de cada um dos alunos.

O uso da tecnologia de informação como um agente de mudança trouxe vários desafios, entre eles está à implementação de técnicas que possibilitem formas de projetar, desenvolver e distribuir material educacional alunos com perfis diferenciados.

A tecnologia dos objetos de aprendizagem tem se mostrado robusta, prática, econômica e uma forte candidata a liderar as alternativas de escolha, devido ao seu potencial de reusabilidade, generalidade, adaptabilidade e escalabilidade.

Assim, um objeto de informação (vídeos, sons, imagens, textos, sites na web, entre outros) quando é utilizado com um objetivo educacional é, na verdade um objeto de aprendizagem, uma vez que é concebido para apoiar um processo de aprendizagem (JOHNSON, 2003). Os objetos de aprendizagem são quaisquer recursos que possam ajudar no processo de ensino e serem reutilizados em diversos contextos (TAROUCO; FABRE; TAMUSIUMAS, 2003). A sugestão de diversificação de conteúdos em diferentes contextos vem sendo tratada com a utilização de SR (Sistemas de Recomendação) (GARCÍA et al. 2009), (GLUZ; VICARI, 2010), (ZHUHADAR et al. 2009), (NERY et al. 2011).

Neste cenário, uma classe de sistemas, chamados de SSCs (Sistemas Sensíveis ao Contexto), utilizam informações extraídas do contexto para fornecer serviços adaptados e relevantes na realização das tarefas do usuário. Dey e Abowd (1999) definem contexto como qualquer informação relevante que possa ser utilizada para caracterizar entidades de uma interação usuário computador, por exemplo, identidade e localização. Segundo Moore et al. (2009) ambientes sensíveis ao contexto se adequam ao perfil do usuário, levando em consideração informações fornecidas pelo próprio usuário, além daquelas captadas dinamicamente a partir de sua interação com os dispositivos computacionais. Chen e Kotz (2000) definem contexto em função de seu efeito sobre uma aplicação, são os conjuntos de estados do meio ambiente que determinam o comportamento de uma aplicação ou causam a ocorrência de um evento específico da aplicação, que é relevante para o usuário. Neste trabalho é utilizada esta definição.

Sistemas sensíveis ao contexto necessitam, em geral, coletar uma grande e diversificada quantidade de informações, analisar essas informações como variáveis independentes, ou combiná-las com outras informações do passado ou presente. Em seguida, transmitem sua interpretação a aplicações que realizam análises. Todo esse processo deve ser repetido de forma automática e adaptado às mudanças. No contexto deste trabalho a questão que se coloca é como, quando e de que forma podemos fazer a recomendação de objetos de aprendizagem que possibilitem aos estudantes: construir conhecimento relacionado a determinados assuntos, desenvolver habilidades precisas relacionadas aos conteúdos tratados, desenvolver nos alunos uma consciência crítica com relação à importância desta competência para que compreendam como e quando utilizá-la.

O projeto BROAD busca adoção de tecnologias de software como ontologias, serviços web semânticos, agentes e *workflow* para o desenvolvimento de ambientes de recomendação personalizada de objetos. Em trabalhos anteriores (BRAGA et al. 2011) (CAMPOS et al. 2012a), (CAMPOS et al., 2011a), (CAMPOS et al. 2012b), (CAMPOS et al., 2011b) e

(SOUZA et al., 2010) o grupo avançou em questões de busca semântica de objetos de aprendizagem, sequenciamento e composição desses objetos.

1.1 METODOLOGIA

A metodologia de pesquisa incluiu a revisão bibliográfica, a proposta de uma arquitetura capaz de realizar recomendação de objetos de aprendizagem sensível ao contexto, baseado em uma ontologia para modelagem do perfil e contexto do aluno em um ambiente *e-learning* e implementado em um sistema multi-agentes. O professor tem papel fundamental no modelo de recomendação ao identificar características dos objetos de aprendizagem que comporão o plano didático e as tarefas, objetivos e competências a serem alcançadas.

Na revisão bibliográfica foram revistos os conceitos, metodologias, métodos e ferramentas que permitiram caracterizar objetos de aprendizagem e seus padrões de metadados, sistemas de recomendação, ambientes sensíveis ao contexto, modelos de alunos e sistemas baseados em agentes.

Após essa caracterização foi feita a proposta da arquitetura BROAD-RS (*BROAD Recommendation System*) e sua avaliação através de cenários. Foi desenvolvida a ontologia PERSONNA para o suporte semântico do perfil e contexto do aluno.

Para definição dos contextos de aprendizagem relevantes neste estudo, utiliza-se uma abordagem apoiada por cenários de aplicação (FLEISCHMANN, 2012). Um cenário de aplicação consiste da descrição completa de uma atividade contextualizada do aluno, onde os valores esperados no início e no fim de uma atividade são claramente definidos.

Cenários são muito explorados pela Web Semântica na demonstração da operação prática de sistemas. Um cenário de aprendizagem típico consiste de diferentes situações internas sendo iniciadas e finalizadas através de eventos, resultado de ações do aluno sobre o ambiente educacional ou vindas do próprio ambiente. O cenário pré-entende a sensibilidade à situação de aprendizagem do aluno por parte do ambiente, mostrando quais elementos contextuais devem ser analisados a cada momento e as possíveis recomendações realizadas (FLEISCHMANN, 2012).

1.2 HIPÓTESES

Uma hipótese posta para este trabalho é a possibilidade de modelar formalmente e implementar um sistema de recomendação de OA de forma a aperfeiçoar as características de adaptabilidade do mesmo ao perfil e contexto do aluno, a partir de plano didático do professor.

Esta hipótese tem origem na própria definição dos ambientes educacionais sensíveis ao contexto, os quais aplicam métodos e técnicas específicas de forma a adaptar o ambiente a fatores relevantes a determinado grupo ou indivíduo que o acessa, buscando sua personalização.

Uma hipótese secundária é a identificação dos dados relevantes para definição da recomendação de objetos de aprendizagem, considerando a adaptação desse ambientes aos modelos e contextos dos alunos.

Outra hipótese secundária é se o uso de agentes na arquitetura de recomendação de OA contribui para a eficácia da adaptabilidade das mesmas ao contexto e perfil dos alunos.

1.3 OBJETIVOS

O objetivo deste trabalho é propor a arquitetura BROAD-RS que permitirá a automação do processo de recomendação de um OA de acordo com o plano didático especificado pelo professor e sensível ao contexto do aluno. Esta proposta facilitará a modularidade, flexibilidade e a escalabilidade do ambiente de aprendizagem, bem como a reutilização de objetos de aprendizagem.

1.4 ORGANIZAÇÃO

Para atingir o objetivo proposto, este trabalho está organizado da seguinte forma: este capítulo apresenta uma introdução as justificativas do problema a ser tratado, a metodologia, hipóteses, os objetivos e a organização da monografia. No capítulo 2, 3, 4 são apresentados os pressupostos teóricos descrevendo as principais tecnologias envolvidas na proposta, explicando o que são objetos de aprendizagem, a recomendação desses objetos de aprendizagem, a análise de contexto, os principais metadados, agentes, e o uso de semântica na recomentação de objetos de aprendizagem. No capitulo 5 são apresentados os trabalhos relacionados, suas análises e comparações. No capitulo 6 é apresentado o projeto BROAD-RS, sua arquitetura de recomendação de OAs, a modelagem dos agentes da arquitetura, a ontologia PERSONNA e cenários de aprendizagem. Ao final são apresentadas as considerações finais e as referências bibliográficas.

2 OBJETOS DE APRENDIZAGEM

O paradigma da orientação a objeto, utilizado na programação de computadores serviu de base para os OAs (Objetos de Aprendizagem) ou LO (*Learning Objects)* em inglês. Conforme Tarouco, Fabre, Ramusiunas (2003), os OA são quaisquer recursos que possam ajudar no processo de ensino e serem reutilizados em diversos contextos.

Professores e projetistas de conteúdo podem construir pequenos componentes que podem ser utilizados inúmeras vezes e em diversos contextos de aprendizagem. Objetos de aprendizagem são distribuídos através da internet e intranets, podendo ser disponibilizados simultaneamente para um considerável número de pessoas.

O grupo de trabalho *Learning Object Metadata Working Group* do IEEE (IEEE, 2012) define como objeto de aprendizagem "qualquer entidade, digital ou não digital, que pode ser usada, reutilizada ou referenciada durante a aprendizagem apoiada por tecnologia". Apesar de o conceito fazer referência a entidades não digitais, este trabalho tem como foco o contexto digital.

Johnson (2003) propõe a anatomia de um OA (Figura 1) e explica que este nome "objetos" vem de duas áreas da prática profissional: a "programação orientada a objetos" e "objetivos de aprendizagem".

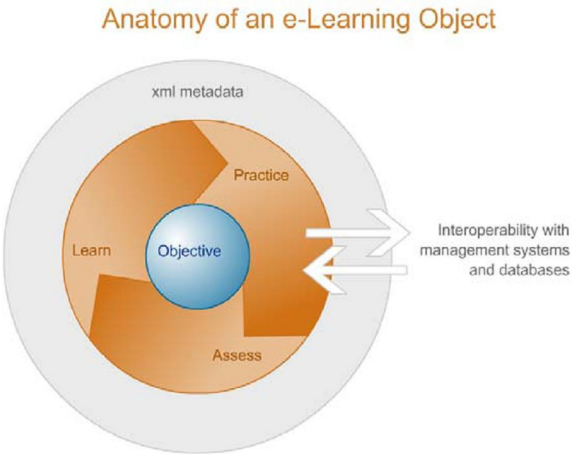


Figura 1: Anatomia de um Objeto de Aprendizagem (JOHNSTON, 2003)

Johnson (2003) define objeto de aprendizagem como uma "coleção de materiais digitais (figuras, documentos, simulações, entre outras) que são acoplados com objetivos de aprendizagem claros e mensuráveis ou projetados para apoiar o processo de aprendizagem". Neste sentido, um artigo científico, uma página web, um simulador, um programa de perguntas e respostas programadas, todos são considerados aqui objetos de aprendizagem.

Johnson (2003) ainda destaca um ponto importante muitas vezes esquecido na discussão dos objetos de aprendizagem: a análise do público-alvo e analise dos objetivos de um determinado assunto em um determinado contexto.

Kay e Kannack (2008) procuraram destacar o foco pedagógico na definição proposta para objetos de aprendizagem, caracterizando-os como “ferramentas interativas baseadas na web que apoiam a aprendizagem incentivando, ampliando e guiando o processo cognitivo dos aprendizes”.

Behar (2009) define objetos de aprendizagem como recursos autônomos, utilizados como módulos de um determinado conteúdo e que podem ser incorporados a múltiplos aplicativos e adaptados a diversos contextos.

Um OA tem como função atuar como recurso didático interativo, abrangendo um determinado segmento de uma disciplina e agrupando diversos tipos de dados e tudo o que pode auxiliar o processo de aprendizagem (Figura 2). Pode ser utilizado tanto no ambiente de aula presencial, quanto na educação a distância.

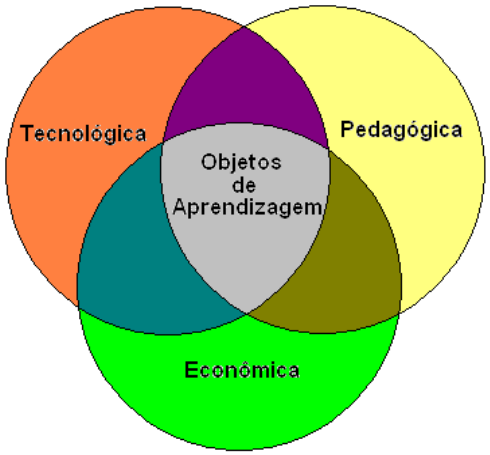


Figura 2: Relação dos objetos de aprendizagem com as três grandes áreas envolvidas em sistemas de *e-learning*.

O uso de OAs para apoiar o processo de aprendizagem é considerado um fator chave para a implantação de uma estrutura *e-learning*. Embora sua flexibilização e reutilização ofereça muitas vantagens, o uso de OAs na preparação de cursos tem duas exigências fundamentais. A primeira diz respeito à identificação adequada dos objetos de aprendizagem e a segunda diz respeito à sua integração em um programa de aprendizagem adequado. Este papel está relacionado com o uso de metadados e perfis dos alunos (TUGSGEREL et al. 2010). Neste contexto, o processo de aprendizagem é mediado por um LMS (Sistema Gestor de Aprendizagem) que, através de um registro, acessa os metadados dos objetos de

aprendizagem publicados por provedores e atua como um repositório de metadados (TUGSGEREL et al. 2010).

Com isto observa-se que os objetos de aprendizagem podem suprir, se bem elaborados e padronizados, necessidades das áreas econômica a qual abrange os fatores relacionados a custos no desenvolvimento destes ambientes, da área tecnológica pois garantem acessibilidade, reusabilidade, interoperabilidade entre outras vantagens que são muitos importantes para o dinâmico desenvolvimento destes ambientes, e também as necessidades da área pedagógica, pois permitem que os aprendizes destes sistemas possam construir seu conhecimento e interagirem mais com os sistemas computacionais.

Conforme Wiley (2001) os objetos de aprendizagem possuem várias características marcantes, as quais justificam sua importância no desenvolvimento de ambientes de aprendizagem, algumas delas são: interoperabilidade, reusabilidade, gerenciabilidade, granularidade, acessibilidade e durabilidade. Abaixo estão destacadas algumas das características dos objetos de aprendizagem:

- **autonomia**: capacidade de serem utilizados individualmente, desvinculados de outros;
- **interatividade**: apresentação do conteúdo de forma dinâmica, permitindo a interação;
- **reutilização em diversos contextos e propósitos**: possibilidade de serem utilizados tanto para seu contexto inicial quanto reutilizados em outras áreas de ensino;
- **agrupamento em conjuntos maiores**: agrupamento de diversos módulos de conteúdos específicos, de maneira a garantir um ambiente de aprendizado mais rico e flexível.

Com base nesses conceitos pode-se tratar os objetos como elementos preparados a fim de dar apoio ao ensino delimitando de maneira clara os tipos de recursos que eles podem utilizar. Os objetos devem ser concebidos com a premissa de serem facilitadores na aprendizagem, e devem servir como instrumento para que o aluno construa o seu entendimento sobre o assunto que está sendo abordado, permitindo a composição de conteúdos (MORALES; AGÜERA, 2012) (Figura 3).

Os OAs devem ser projetados de forma a integrar-se com outros semelhantes e que esta junção possa possibilitar a criação de conteúdos maiores. Ou seja, eles são módulos que tem o objetivo de se ligar e possibilitar a construção do conhecimento (HODGINS;

CONNER, 2001).

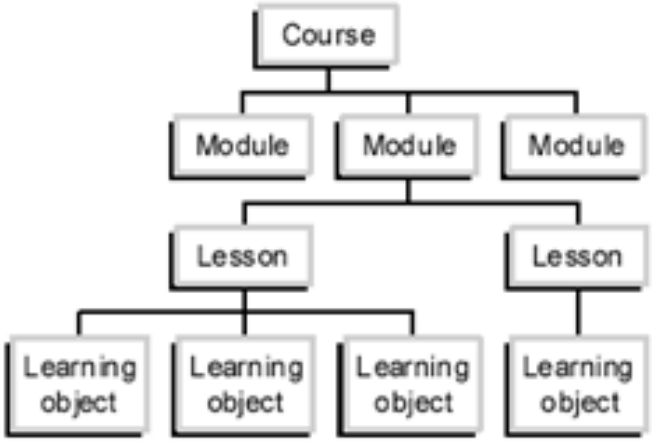


Figura 3: Estrutura de um curso (MORALES; AGÜERA, 2012)

Hodgins (2001) apresenta uma análise taxonômica de OA apresentada na Figura 4.

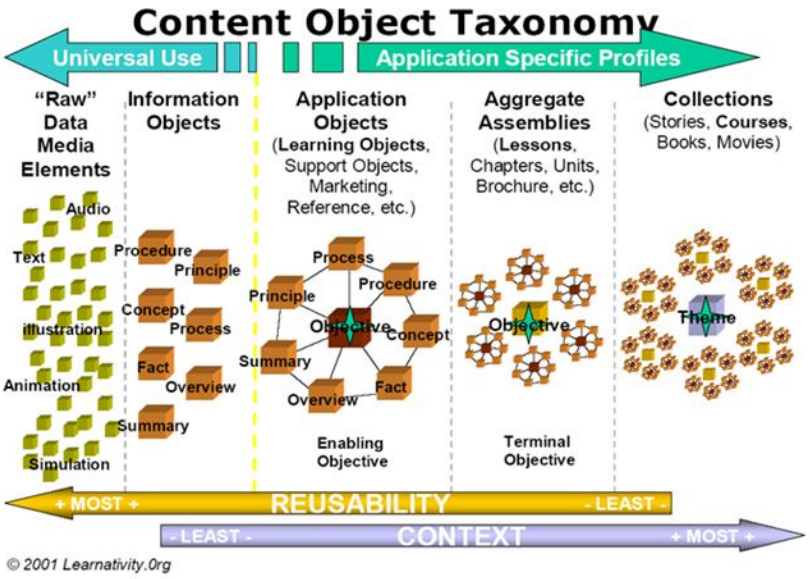


Figura 4: Taxonomia de Objetos de Aprendizagem (HODGINS, 2001)

Note-se que esta hierarquia genérica pode ser usada em várias aplicações e que os dois primeiros níveis são referentes a recursos digitais e, em seguida, Hodgins (2001) específica camadas para aplicações, tais como os objetos de aprendizagem. Dados ou elementos brutos de mídia são o menor nível nessa taxonomia, ou seja, elementos "puros" como imagens, textos, áudios, vídeos, exercícios, e tudo o que pode auxiliar o processo de aprendizagem. No segundo nível temos os blocos de dados, objetos de informações organizados em bloco e no terceiro nível desta taxonomia temos os próprios objetos de aprendizagem. O quarto e quinto níveis são definidos com o objetivo de se criar conjuntos agregados, como lições ou capítulos, o que pode por sua vez auxiliar na montagem de coleções cada vez maiores, como cursos e

currículos completos.

O poder da flexibilidade e reutilização dessa taxonomia de conteúdo torna-se clara (HODGINS; 2001). Esta grande massa de dados digitais pode ser armazenada em um repositório de banco de dados e com a ajuda de metadados para detalhar e descrever seus atributos, cada um está pronto para ser usado e reutilizado a partir de composições ou individualmente, dentro de diversos contextos e aplicações.

A idéia central do conceito de OAs é permitir que os projetistas educacionais construam componentes educativos relativamente pequenos que possam ser usados em diferentes contextos de aprendizagem, ou seja, são conteúdos que, além de permitir alcançar um objetivo educacional, promovem a reusabilidade dos conteúdos utilizados no processo de ensino e aprendizagem.

Colocando de uma forma mais simples, um OA, é um pequeno “pedaço” de conteúdo voltado para um objetivo de aprendizado específico. Estes objetos podem conter um ou mais componentes, os quais incluem textos, imagens, vídeos, animações, exercícios ou similares. A reutilização será suportada tanto em nível de objetos de aprendizagem como em nível de componentes, e devido à maneira padronizada na qual estes objetos serão construídos e indexados, serão simples de achar e usar.

2.1 ASPECTOS PEDAGÓGICOS

A utilização de OAs possibilita ao aluno conhecer novas ferramentas, testar diferentes situações, arriscar, antecipar, compreender a relação entre causa e efeito de conceitos e fenômenos, para despertar a curiosidade e para resolver problemas.

Para que um objeto de aprendizagem tenha eficácia didática, segundo Borges e Navarro (2005), o educador deve procurar responder algumas questões: Quais os padrões e ferramentas aplicadas na construção do objeto de aprendizagem? Qual será o seu objetivo? O que ele vai ou não abordar? Com que profundidade? Qual enfoque adequado? Para que público? Qual a importância deste tópico para o conhecimento (disciplina e/ou curso) que se deseja transmitir? Quais formas de interatividade com o educando?

Um objeto de aprendizagem pode ser planejado de modo a assegurar o desenvolvimento gradativo das competências e habilidades do indivíduo. Este tipo de abordagem vislumbra o objeto de aprendizagem trabalhando de forma colaborativa com outros recursos educacionais.

O uso das tecnologias da informação como instrumento para construção do conhecimento está em um processo de forte expansão, justamente por possibilitar às escolas a

realização de experiências além da sala de aula. Entretanto, a utilização da tecnologia no ensino não deve ser feita de maneira ingênua e prematura, mas sim acompanhada de um estudo abrangente sobre como um sujeito adquire e constrói o conhecimento (ZANETTE et al. 2006).

O processo de aprendizagem agrega três fatores importantes: que o conteúdo seja estruturado de maneira compreensível, que o aluno tenha um conhecimento prévio que lhe permita entender o que será abordado e confiança do aluno em aceitar este tipo de ensino relacionando-o com o que ele já conhece previamente. Portanto, um objeto de aprendizagem pode ter sucesso no intuito de auxiliar na construção do conhecimento se for capaz de servir de ponte entre o conhecimento básico, que o aluno já possui, e o conhecimento avançado que se pretende agregar. Este objetivo pode ser alcançado com o uso de recursos visuais como animações e simulações que despertem a atenção do aluno e façam com que ele possa vivenciar o tema abordado (BORGES; NAVARRO, 2005).

Observa-se que a utilização dos objetos de aprendizagem pode prover ao aluno um apoio na construção de conhecimentos e conceitos de maneira mais interativa e eficiente. Não há um modelo pedagógico único, uma fórmula pronta de se apresentar um conteúdo por meio de um OA.

Constata-se que a construção e a utilização de um objeto de aprendizagem dependem do planejamento do educador de modo a possibilitar a construção do conhecimento com o apoio de um conteúdo interativo e atraente.

2.2 PADRÕES DE OBJETOS DE APRENDIZAGEM

Os padrões de OAs constituem um meio de organizar os dados de um OA para prover comunicação entre diferentes ambientes computacionais (interoperabilidade), bem como garantir seu acesso e usabilidade. Embora tenha surgido na década de 60, só 20 anos depois o termo “metadados” começou a aparecer com mais frequência na literatura especializada. Metadados são definidos como dados sobre dados, isto é, metadado é a informação sobre o dado que permite seu acesso e gerenciamento de maneira eficiente e inteligente (TAYLOR, 2003).

Segundo Hodge (2001) metadados podem ser definidos como informações estruturadas que descrevem, explanam, localizam ou facilitam a recuperação, o uso e o gerenciamento de um recurso de informação.

Usualmente, os OAs são formados por diversos elementos (vídeo, texto, animação, áudio, entre outros) que interagem entre si e o usuário e, para reutilizá-los, é preciso descrevê-

los. Dessa forma, é necessário utilizar metadados que tratem dos elementos que compõem um OA, ou seja, ampliar o alcance dos metadados para tratamento dos elementos, não apenas a descrição geral do OA.

Segundo Campos et al. (2011b) existem diversificados metadados a serem cadastrados sobre um OA e por isso, vários padrões têm surgido para tornar os objetos mais acessíveis. Várias organizações criaram padrões, a seguir apresentamos alguns deles.

2.2.1 PADRÃO SCORM

Em 1977 o departamento de defesa americano estabeleceu o ADL *(Advanced Distributed Learning Initiative)* com o objetivo de desenvolver uma estratégia para o uso de tecnologia de informação e aprendizagem para modernizar a educação e treinamento, bem como promover uma estreita cooperação entre governo, academia e a indústria para estabelecer uma padronização de componentes que pudessem ser usados nos ambientes virtuais de aprendizado. Desta iniciativa resultou a definição de uma série de especificações e requisitos de alto nível para objetos de aprendizagem, que promovesse conteúdo reusável, acessibilidade, durabilidade e interoperabilidade. Ao conjunto destas especificações e requisitos chamou-se SCORM (*Sharable Content Object Reference Model)* (SCORM, 2004).

No padrão SCORM (2004), os metadados podem ser contexto independente ou contexto dependente. Metadados contexto independentes referem-se normalmente a metadados imutáveis que descrevem objetos de conteúdo, por exemplo. Metadados contexto dependentes dizem respeito à metadados que fazem sentido somente no contexto de uma dada organização. O padrão SCORM incorporou dados do ARIADNE (2002) e do IMS (2002), entre outros.

SCORM é composto de três subespecificações: a seção de empacotamento de conteúdo especificando como o conteúdo deve ser empacotado e descrito; a seção de *run-time* especifica como o conteúdo deveria ser lançado e como se comunica com o sistema de gerenciamento da aprendizagem e a seção de sequenciamento especifica como o aluno navega entre as partes do curso. A Figura 5 ilustra a organização do SCORM como um conjunto de especificações de outras organizações contidas ou referenciadas no modelo.

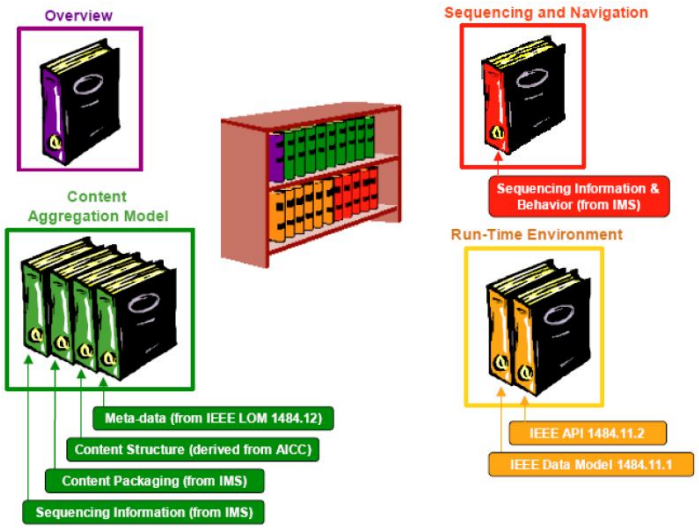


Figura 5: SCORM como conjunto de especificações (DUTRA; TAROUCO, 2006)

2.2.2 PADRÃO LOM-IEEE

O comitê de padrões de tecnologia para aprendizagem IEEE (2012) foi instituído pelo IEEE, pela *Computer Society* e pelo *Standards Activity Board* com o objetivo de desenvolver normas, orientações e práticas recomendadas na área de aprendizagem apoiada por computador (HODGINS, 2002). Os cerca de 20 grupos de estudos constituídos no LTSC *(IEEE Learning Technology Standards Committee*) classificam-se nas categorias de atividades gerais, atividades relacionadas ao aprendiz, atividades relacionadas ao conteúdo, dados e metadados, sistemas de gerência e aplicações. Esse conjunto de metadados para OAs tem suas origens nos projetos ARIADNE (2002) e IMS (2002) e também se baseia em estudos de metadados desenvolvidos pelo grupo do Dublin Core (2001).

Segundo o IEEE (2012), o padrão LOM (*Learning Object Metadata)* foi desenvolvido pelo LTSC *(IEEE Learning Technology Standards Committee*) e tem como objetivo facilitar a busca, a avaliação e o uso de OAs, além de facilitar a acessibilidade e a interoperabilidade. Para cada elemento de dados, o esquema de base define os campos descritos na Tabela 1.

Tabela 1: Conjunto mínimo de atributos necessários LOM (IEEE, 2012) e (CAMPOS et al. 2011b)

Metadados	Descrição
Nome	O nome pelo qual o elemento de dado é referenciado
Explicação	A definição do elemento de dados
Tamanho	O número de valores permitidos
Ordem	Se a ordem dos valores é significativa (aplicável apenas para os elementos de dados com múltiplos valores)
Espaço de valor	O conjunto de valores permitidos para o elemento de dados normalmente sob a forma de um vocabulário ou uma referência à outra norma (como vCard, ISO8601 (VCARD, 1996) para a representação de datas, entre outros)

Tipo de dados	Um conjunto de valores distintos
Exemplo	Um exemplo ilustrativo

O LOM é um modelo que busca especificar a estrutura e a semântica dos metadados de OAs, definindo os atributos necessários para sua descrição. O conjunto de metadados proposto no LOM busca definir o mínimo de atributos necessários para permitir que os OAs sejam gerenciados, avaliados e localizados, sendo assim pode e deve ser extendido. Alguns atributos relevantes que podem ser citados são: tipo do objeto, autor, proprietário, termos de distribuição e formato, entre outros. Além dos atributos tradicionais relacionados com a descrição do objeto, também existem atributos pedagógicos: nível do conteúdo, pré-requisitos, estilo de interação, entre outros (PEREIRA; MELO, 2003). A Figura 6 ilustra a estrutura de metadados do LOM.

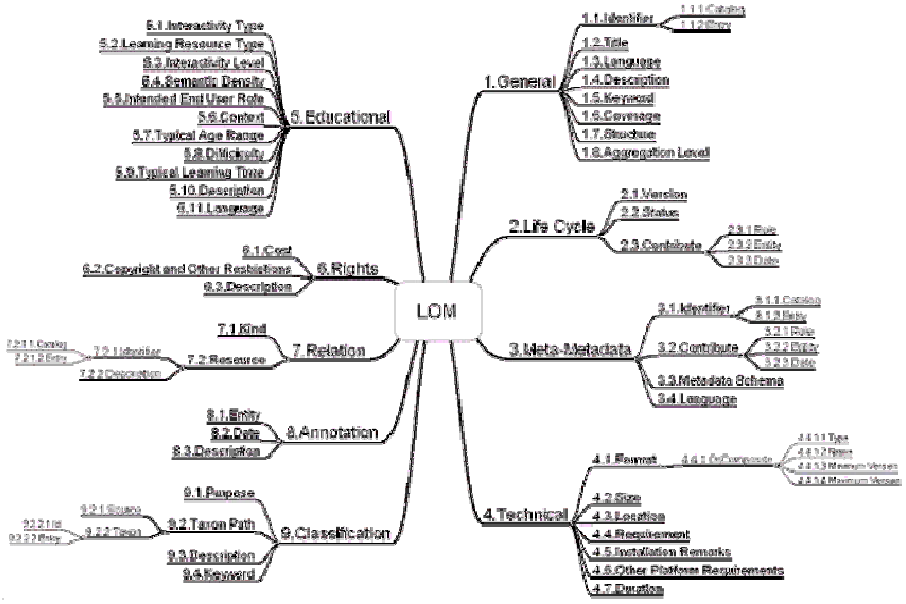


Figura 6: Elementos e estrutura do esquema conceitual de dados do LOM extraído de (http://www.imsglobal.org/metadata/mdv1p3pd/imsmd_bestv1p3pd.html)

O padrão especifica um esquema conceitual de dados, que define a estrutura da instância de metadados de um objeto de aprendizagem, descrevendo as características relevantes do OA ao qual se aplica. Esses elementos de dados formam uma hierarquia (nós intermediários ou agregações e folhas), onde o primeiro nível compõe-se das seguintes categorias:

1. **características gerais**, que agrupa as informações gerais que descrevem o OA como um todo;
2. **ciclo de vida**, que agrupa as características relacionadas à história e estado corrente desse OA e aquelas que afetaram esse OA durante sua evolução;

3. **meta-metadados**, agrupa as informações sobre a instância de metadados propriamente dita (ao invés do conteúdo sendo descrito);
4. **técnicas**, agrupa as características e requisitos técnicos do OA;
5. **educacionais**, agrupa as características educacionais e pedagógicas do OA;
6. **direitos**, agrupa os direitos de propriedade intelectual e condições de uso do OA;
7. **relações**, agrupa as características que dizem respeito às possíveis relações entre o OA e outros OAs;
8. **anotação**, agrupando os elementos que contêm comentários sobre o uso educacional do OA e sobre por quem e quando os comentários foram feitos;
9. **classificação**, que agrupa os elementos que descrevem o OA com relação a um sistema de classificação específico.

Em função da completitude do padrão LOM (IEEE, 2012), da importância do IEEE como organização de padronização e da preocupação dos órgãos que se dedicaram aos seus desenvolvimentos em documentá-los, é adotado o mesmo como um dos padrões da arquitetura BROAD-RS.

2.2.3 PADRÃO OBAA

O padrão OBAA (Objetos de Aprendizagem Baseados em Agentes) foi proposto em julho de 2009, pela FAURGS (Fundação de Apoio da Universidade Federal do Rio Grande do Sul). A interoperabilidade de OAs, no contexto da utilização destes objetos nas plataformas de TV Digital, *Web* e dispositivos móveis, é o principal critério adotado pelo projeto OBAA para a análise dos padrões e tecnologias atuais de OAs e para a especificação de novos padrões. O objetivo é criar um padrão de metadados compatível com o padrão LOM, mas que permita a interoperabilidade dos OAs nas plataformas Web, TV Digital e dispositivos móveis (Figura 7), além de suportar requisitos de acessibilidade para pessoas com necessidades especiais e registrar informações educacionais específicos do contexto brasileiro (OBAA, 2010).

Figura 7: OAs nas plataformas Web, TV Digital e dispositivos móveis (VICARI et al. 2009)

No padrão OBAA foi proposta uma extensão da categoria *Educational* do LOM (Figura 8). Ela ampliou a quantidade de elementos desta categoria, tendo como base o modelo interacionista.

Figura 8: Categoria Educational (VICARI et al. 2009)

Para lidar com as questões de acessibilidade, foi criada uma nova categoria no padrão OBAA denominada *Acessibility* (Figura 9). Este tem por intenção armazenar informações de acessibilidade sobre o estudante, para definir as configurações dos usuários ao acessar o OA. Estas informações permitem atender, por exemplo, aos requisitos de áudio para cegos, legenda para surdos, idiomas e outras especificações importantes para dar acesso e inclusão ao OA às pessoas com necessidades educativas especiais (VICARI et al. 2009).

Pedagogicamente, é importante definir a interação entre o OA e o usuário. A interação diz respeito ao comportamento das pessoas em relação a outras pessoas e sistemas.

Figura 9: Categorias *Acessibility* (VICARI et al. 2009)

Provavelmente, somente é possível proporcionar experiências de aprendizagem significativas se a solução educacional for projetada interativa. Isso significa disponibilizar uma interface que possibilite interação e, mais do que isso, oferecer atividades de aprendizagem que exijam do aluno interação com conteúdos, ferramentas e com outras

pessoas. Para tanto, a definição de interação foi contemplada com a especificação do metadado *interaction*, composto pelo mecanismo sensorial utilizado para transmitir a informação, pela forma de interação entre o usuário e o objeto, mecanismos para informar e utilizar a copresença de outros usuários no ambiente e pela forma de relacionamento entre os usuários, necessária para o funcionamento deste objeto de aprendizagem (VICARI et al. 2009; BEHR et al. 2012).

Para facilitar o uso do OBAA, foram definidos conjuntos de metadados direcionados aos desenvolvedores de conteúdo educacional. Através destes metadados é possível atender diferentes demandas de nichos específicos da comunidade educacional, mantendo a aderência ao padrão e reduzindo sua complexidade.

2.2.4 PADRÃO BROAD

Segundo Campos et al. (2011b), a seleção dos metadados para o projeto BROAD iniciou-se pela verificação de quais os metadados estão mais presentes nos padrões e os mais usados para catalogação de OAs em repositórios (Tabela 2).

Tabela 2: Metadados BROAD (CAMPOS et al. 2011b)

Metadados	Descrição
Geral	Características gerais do OA
Ciclo de Vida	Características referentes à atual situação do OA
Técnico	Características técnicas do OA
Educacional	Características educacionais do OA
Direitos	Características de direitos autorais do OA
Classificação	Itens que classificam o OA
Acessibilidade	Características referentes à acessibilidade do OA

O trabalho foi desenvolvido em três etapas: na primeira etapa foram selecionados os metadados considerados essenciais, numa segunda etapa foram definidos metadados educacionais e na terceira e última etapa foi selecionado um conjunto de características de qualidade para os OA a serem validados

O conjunto de metadados do projeto BROAD inclui um conjunto de elementos de metadados, centrando-se em critérios de aprendizagem. Os metadados definidos nesse projeto visam garantir mais do que os metadados educacionais tradicionais, mas também uma lista de verificação com informações sobre o cenário de uso desses OA. Esses metadados são descritos através de vários sub-metadados, representados num primeiro nível de expansão na Figura 10.

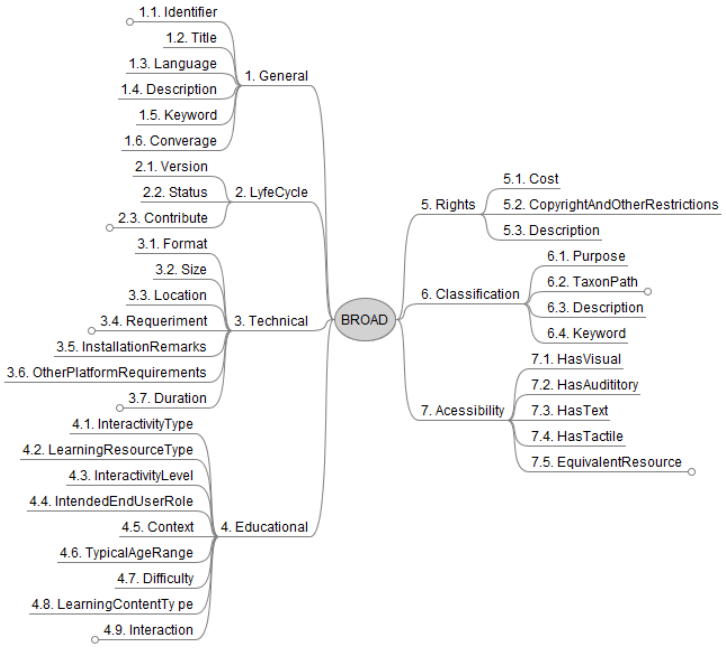


Figura 10: Metadados essenciais BROAD no segundo nível de detalhamento

O projeto BROAD estende os metadados Educacionais e os descreve através de sub-metadados constantes na Tabela 3. Esta categoria permite a descrição das características educacionais e é tipicamente usada por professores, gestores, autores e alunos.

Tabela 3: Metadados Educacionais BROAD

Nome	**Descrição**	**Opções**
1.Tipo interatividade	Nível de interatividade	Ativo; expositivo; misto
2. Tipo	Descreve o tipo do arquivo	Exercício; simulação; questionário; diagrama; figura; gráfico; apresentação; planilha; texto; avaliação; experimento; problema; autoavaliação; conferência
3. Nível interatividade	A interatividade neste contexto se refere ao grau em que o aluno pode influenciar o aspecto ou o comportamento do componente	Muito baixo; baixo; médio; alto; muito alto
4. Usuário final	Representa o principal usuário(s) para que o objeto foi projetado	Professor; autor; aluno; gestor
5. Contexto	Foco em qual o objeto se aplica	Fundamental; médio; superior; treinamento; outro
6. Média de idade	Intervalo de idade	
7. Dificuldade	Nível de dificuldade	Muito fácil; fácil; médio; difícil; muito difícil
8. Tipo conteúdo	Especificação educacional do tipo do conteúdo do objeto de aprendizagem	Factual; conceitual; procedimental; atitudinal
9. Interação	Especifica a interação educacional proposta por este objeto de aprendizagem e seu(s) usuário(s)	
9.1. Percepção	Especifica a forma sensorial pela qual o aluno receberá as informações do objeto de aprendizagem	Visual; auditiva; mista; outra
9.1.2. Co-presença	Especifica a utilização de mecanismos que auxiliem a identificação de outros usuários no	

	ambiente	
9.1.3. Reciprocidade	Forma de relacionamento entre os usuários, necessária para o funcionamento deste objeto de aprendizagem	

2.3 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Objetos de aprendizagem representam a possibilidade de estruturar um curso ou disciplina com uma grande variedade de recursos e mídias. Para que os mesmos atendam os objetivos educacionais é necessário que estejam disponíveis para reuso e adaptação. Isso é feito através do cadastro e busca por metadados.

Analisando os padrões de metadados disponíveis para objetos de aprendizagem, observa-se que os padrões LOM e OBAA são bastante semelhantes. O padrão OBAA usa onze categorias, sendo que destas, nove são coincidentes com as categorias propostas pelo LOM. O OBAA também cria dois novos metadados: acessibilidade e segmento e estende o esquema de dados das categorias requisitos técnicos e características educacionais. Já o padrão BROAD, (CAMPOS et al. 2011b), baseia-se nos padrões LOM e OBAA, mas destaca as características educacionais.

Assim, visando garantir mais do que os metadados tradicionais, a arquitetura BROAD-RS adota os metadados do padrão BROAD para cadastro dos OAs e algumas características dos padrões BROAD (CAMPOS et al. 2011b), LOM (IEEE, 2012) e o OBAA (VICARI et al. 2009) para definição de seu plano didático.

3 RECOMENDAÇÃO, ONTOLOGIAS E MODELOS DE ALUNOS

Este capítulo faz uma revisão bibliográfica sobre sistemas de recomendação (os tipos de técnicas de filtragens, contexto, sensibilidade ao contexto, ambientes sensíveis ao contexto), rede de ontologias e padrões e modelos de alunos.

3.1 SISTEMAS DE RECOMENDAÇÃO

Geralmente muitos AVAs (Ambientes Virtuais de Aprendizagem) enfocam os aspectos relacionados à sua funcionalidade esquecendo da função pedagógica do ambiente. Este desafio é intensificado quando se busca recomendar objetos de aprendizagem distintos, personalizados para cada aluno, com base em necessidades individuais, competências e interesses.

O foco na recomendação de objetos de aprendizagem voltados ao desenvolvimento de competências exige uma mudança de paradigma educacional, desprendendo-se de modelos pedagógicos fundamentados em currículos e metodologias focadas na educação. Volta-se então para modelos pedagógicos mais flexíveis, respeitando características sociais, interesses, necessidades e limitações de cada indivíduo (CAZELLA et al. 2009).

São encontrados na literatura várias definições para o conceito de competência. Do dicionário Aurélio da língua portuguesa, obtém-se a seguinte definição: “Qualidade de quem é capaz de apreciar e resolver certo assunto, fazer determinada coisa, capacidade, habilidade, aptidão, idoneidade”. Nesta definição, contudo, não ficam explícitas as relações entre os conceitos: habilidades, assuntos, conhecimentos (CAZELLA et al. 2009).

Sugerir conteúdos dentre infinitas possibilidades vem sendo tratado com a utilização de Sistemas de Recomendação. Ferramentas que utilizam dos seus métodos vem sendo amplamente utilizadas em domínios de entretenimento com significativo sucesso.

Existem várias aplicações de recuperação de conteúdo que buscam auxiliar os usuários na identificação de itens de interesse. Porém, é comum que estas aplicações tragam muito conteúdo irrelevante (ADOMAVICIUS; TUZHILIN, 2005). Buscando minimizar este problema, os sistemas de recomendação surgiram, focando na busca por informações relevantes de acordo com características do próprio usuário, bem como em determinados requisitos relacionados aos itens que se quer encontrar.

Segundo Burke (2002) um SR pode ser qualquer sistema que produza recomendações

individuais ao usuário ou que tenha o efeito de guiar o usuário através de recomendações personalizadas a itens que sejam de seu interesse ou que o sejam úteis diante de uma vasta gama de opções. Um sistema de recomendação pode também identificar similaridade entre o comportamento dos usuários e recomendar itens, como objetos de aprendizagem, que já foram recomendados para outros usuários.

Um SR é um sistema que sugere recomendações ao usuário baseados em suas preferências e, para Cazella (2010), auxiliam no aumento da capacidade e eficácia deste processo de indicação já bastante conhecido na relação social entre seres humanos. Como um amigo confiável, o qual recomenda filmes, musicas ou lugares para visitar, os sistemas de recomendação devem fazer sugestões que se adequem as expectativas dos usuários e este é um dos grandes desafios. “Realizar a combinação adequada entre as expectativas dos usuários e os produtos, serviços e pessoas a serem recomendados, ou seja, definir e descobrir este relacionamento de interesses é o grande problema” (CAZELLA, 2010).

Os SR são amplamente usados no âmbito de *e-commerce*, livrarias e mídias sociais. No entanto, a recomendação em ambiente educacional tem suas peculiaridades. A aprendizagem requer um esforço que, muitas vezes, leva mais tempo e mais interações se comparada a uma transação comercial. O processo de recomendação para um aluno não termina quando o produto é adquirido, como acontece em uma compra, ao contrário, os alunos seguem o processo e alcançam diferentes níveis de competências que têm vários estados em diferentes domínios (MANOUSELIS, 2011).

Segundo Burke (2002) e Adomavicius e Tuzhilin (2005) os SRs podem ser classificados em tipos básicos: filtragem colaborativa, baseados em conteúdo e híbridos. Quando se fala em recomendação baseada em filtragem colaborativa o sistema realiza sua recomendação através da comparação dos usuários e a similaridade existente entre eles, ou seja, os objetos que são recomendados são os já utilizados anteriormente por usuários com características similares as de quem deseja a recomendação. Dessa forma, é encontrado um conjunto de usuários cujos gostos são semelhantes aos gostos do usuário em questão, os quais são denominados vizinhos mais próximos, a serem vistos mais adiante. Já a recomendação baseada em conteúdo é realizada através dos itens similares ao que o usuário já utilizou anteriormente. Este tipo de recomendação baseia-se na recuperação de informação e faz uso de técnicas de extração de informação, buscando descobrir informações explícitas, caso venham descritas no metadados, ou implícitas, onde são feitas comparações dos conteúdos utilizados e analisadas suas similaridades. No entanto, em sua maioria os sistemas de recomendação utilizam a junção das duas técnicas, aproveitando a similaridade entre os

usuários e entre os objetos utilizados, atingindo uma gama maior de possibilidade e gerando sugestões de novos objetos que diretamente não parecem estar relacionados.

Duas técnicas largamente utilizadas, principalmente em conjunto, são a do vizinho mais próximo e o agrupamento. O algoritmo do vizinho mais próximo calcula as distâncias entre os elementos com base nas propriedades destes. Basicamente, estas propriedades são valoradas, e a distância entre dois elementos corresponde às diferenças entre os valores atribuídos. Isto faz com que vizinhos que não tenham informações parecidas com o item a ser recomendado sejam ignorados. Quando da sua utilização em grandes bases de dados, o algoritmo do vizinho mais próximo torna-se uma técnica muito lenta. Para contornar esse problema, costuma-se utilizar o algoritmo em questão combinado-o com algum outro que restrinja o volume a ser trabalhado, normalmente o agrupamento (ADOMAVICIUS; TUZHILIN, 2005).

O agrupamento consiste de uma classificação de dados não supervisionada (agrupamento de padrões não rotulados) e com base em alguma forma de semelhança, organizada em grupos de acordo com padrões. Assim, normalmente elementos pertencentes ao mesmo grupo tendem a ser mais similares que elementos de grupos diferentes. Existem problemas de agrupamento em várias áreas do conhecimento, o que dificulta a adoção de uma forma de agrupamento genérica, já que há várias iniciativas diferentes relacionadas à técnica. Como grupos representam parte dos atributos dos elementos, esta técnica é menos precisa que outras como o vizinho mais próximo.

Alguns modelos expandem o modelo de Burke (2002) ou geram pequenas modificações nessa classificação como, por exemplo, a inserção do perfil do usuário como técnica de recomendação (NUNES, 2009).

Hinz et al. (2011) menciona também técnicas de mineração de dados, através da qual regras resultantes do processo de mineração são usadas para a recomendação, e a técnica baseada em casos, que usa a lógica, utilizada na resolução de problemas anteriores, dentro do mesmo domínio de aplicação, com pequenas alterações, para resolver novos problemas.

3.1.1 FILTRAGEM COLABORATIVA

Segundo Burke (2002), a FC (Filtragem Colaborativa) é a técnica mais conhecida e mais implementada. Ela reúne as notas ou recomendações atribuídas pelos usuários aos itens, identifica as similaridades entre os usuários baseadas em suas avaliações e gera novas recomendações com base em comparações entre os usuários. Segundo Lichtnow et al. (2006), define-se que usuários com perfil similar são os usuários que avaliaram os mesmo itens de

forma similar.

Segundo Schafer et al. (1999), esta técnica de recomendação pode ser chamada de recomendação de “pessoa para pessoa”. A filtragem colaborativa possui as vantagens de que o sistema não analisa o conteúdo dos itens avaliados, evitando uma complexidade maior na sua implementação e facilitando a recomendação de itens como documentos e músicas onde a variação nos gostos do usuário influencia na variação das suas preferências, e também pode realizar recomendações inesperadas ao usuário. Porém para que as recomendações sejam relevantes é necessário que haja uma grande quantidade de usuários no sistema e usuários que possuam interesses não muito comuns receberão recomendações pobres.

3.1.2 FILTRAGEM BASEADA EM CONTEÚDO

Segundo Lichtnow et al. (2006) a FC (Filtragem por Conteúdo) parte do princípio de que os usuários tendem a se interessar por itens que se interessaram no passado, tornando-se necessária a identificação de associação entre os itens. Diferente da filtragem colaborativa, esta técnica analisa o conteúdo dos itens.

A recomendação realizada por esta técnica ocorre através da associação entre a similaridade entre o conteúdo dos itens e a análise dos interesses do usuário ou do conteúdo dos itens já avaliados pelo usuário.

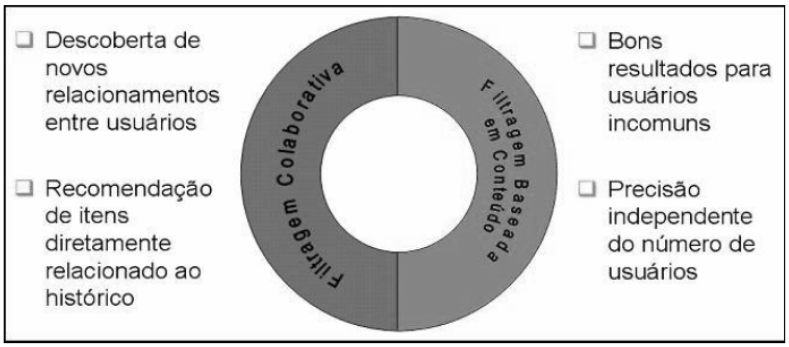
Segundo Schafer et al. (1999), esta técnica pode ser chamada de recomendação de “item para item”, pois baseia-se no conteúdo dos itens. Está técnica não realiza a avaliação qualitativa dos itens que são recomendados, pois basta que haja similaridade entre os atributos que são comparados para que o item seja recomendado para o usuário.

A técnica de filtragem baseada em conteúdo apresenta fragilidade na recomendação de itens com pouca descrição, pois é baseado nesta descrição dos itens que o sistema realiza a recomendação ao usuário. A descrição dos itens deve ser bem elaborada, pois até mesmo a utilização de sinônimos pode gerar recomendações erradas.

3.1.3 FILTRAGEM HÍBRIDA

A abordagem da FHC (Filtragem Híbrida) procura, basicamente, combinar os pontos fortes da filtragem colaborativa e da filtragem baseada em conteúdo visando criar um sistema que possa melhor atender as necessidades do usuário. Essa abordagem é constituída de vantagens proporcionadas pela filtragem baseada em conteúdo e pela filtragem colaborativa, unindo o melhor das duas técnicas e eliminando as fraquezas de cada uma, conforme apresentado pela

Figura 11.



.

Figura 11. Filtragem híbrida (CAZELLA et al. 2010)

3.1.4 FILTRAGEM BASEADA EM MINERAÇÃO DE DADOS

Além das técnicas de filtragem de informação, outras também já foram usadas no contexto dos sistemas de recomendação, com destaque para a mineração de dados, onde as regras associativas extraídas de bases de dados existentes são usadas para a geração das recomendações. Além destas, podemos também citar outras abordagens menos convencionais, como ontologias e lógica *fuzzy*, por exemplo.

A mineração de dados tem por objetivo aplicar técnicas e ferramentas adequadas, a fim de extrair informações padronizadas em um grande volume de dados, que possam ser utilizados tanto para descrever características do passado como para prever situações futuras. Mas, para que a descoberta de conhecimentos seja relevante, é importante estabelecer metas bem definidas que são alcançadas por meio de tarefas de *Data Mining* e consistem na especificação do que estamos querendo buscar nos dados, que tipo de regularidades ou categoria de padrões temos interesse em encontrar, ou que tipo de padrões poderiam nos surpreender (HINZ et al. 2011).

Em relação às técnicas que podem ser usadas em *Data Mining* há uma grande apresentação na literatura, como por exemplo, as técnicas de redes neurais, regras de indução, árvores de decisão, análise de séries temporais, visualização e algoritmos genéticos.

3.1.5 RACIOCÍNIO BASEADO EM CASOS

A RBC (Raciocínio Baseado em Casos) é uma técnica de recomendação que tem origem em estudos de aspectos relacionados ao aprendizado humano, sendo muito utilizada para a resolução de problemas especialistas.

A idéia básica de um sistema RBC é que, assim como acontece na realidade, onde

geralmente usamos as situações passadas para resolvermos algum problema atual, ou seja, para resolver um problema, reutilizam situações anteriores, pois as mesmas tendem a se repetir.

São quatro as etapas principais no desenvolvimento de um sistema de RBC: recuperar, reutilizar, revisar e reter. Estas etapas conduzem e orientam o raciocínio empregado por um sistema de Raciocínio Baseado em Casos que é desempenhado através dos casos (HINZ et al. 2011).

Existem diferentes algoritmos voltados para a recuperação de casos, como por exemplo: *Nearest-neighboor, Kd-tree, Fish-and-Sink, Crash memory model, KDSA (Knowledge-directed Spreading Activation), CRNs (Case Retrieval Nets), OCRNs (Objectdirected Case Retrieval Nets)* (HINZ et al. 2011).

3.2 CONTEXTO

Com o crescimento dos sistemas baseados na Web, cresce o interesse em fazer com que estes sistemas apresentem caráter adaptativo, sendo capazes de realizar análises a respeito do contexto atual de interesse do usuário e se adaptar em atendimento às necessidades particulares deste usuário.

Segundo Moore et al. (2009) o contexto pode ser definido de acordo com informações relativas a propriedades que se combinam para descrever e caracterizar uma entidade e seu papel de uma forma legível pelo computador. Ambientes sensíveis ao contexto se adéquam ao perfil do usuário, levando em consideração informações fornecidas pelo próprio usuário, além daquelas captadas dinamicamente a partir de sua interação com os dispositivos computacionais.

Para Vieira et al. (2009):

> "contexto é o que está por trás da habilidade de discriminar o que é ou não importante em um dado momento e, com isso, permite ajudar indivíduos a melhorar a qualidade de conversação e a compreender certas situações, ações ou eventos".

Segundo Fleischmann (2012), de forma simples, pode-se entender que contexto se trata de tudo o que ocorre ao redor do usuário e que influencia na forma com que este interage com o ambiente físico e com as outras pessoas.

3.2.1 SENSIBILIDADE AO CONTEXTO

Segundo Fleischmann (2012), a sensibilidade ao contexto busca analisar e descrever o comportamento de um sistema de acordo com as mudanças que ocorrem em seu interior. Sistemas sensíveis ao contexto se dispõem a apresentar caráter proativo às modificações ocorridas, adaptando- se a elas. A tarefa de adaptação é bastante complexa, uma vez que para determinar as alterações no contexto do ambiente muitas vezes não são avaliadas modificações isoladas, mas de vários aspectos combinados.

O termo sensibilidade, em inglês *awareness*, trata da capacidade de um sistema de perceber ou estar ciente da ocorrência de eventos, objetos ou padrões de sensores existentes ao redor. Do ponto de vista das interações entre agentes e o ambiente, é definido simplesmente como o conhecimento a respeito do estado de algum ambiente, limitado no tempo e no espaço . O termo sensibilidade usado isoladamente ou em conjunto com contexto possui basicamente o mesmo significado, como será constatado após a definição de contexto.

Além de contexto, Santos et al. (2008a) determina o conceito de elemento contextual, definido como qualquer dado, informação ou conhecimento que permite caracterizar uma entidade em um domínio. Relaciona ainda o elemento contextual com o contexto, determinando que o contexto da interação entre um agente e uma aplicação, para executar alguma tarefa, consiste do conjunto de elementos contextuais instanciados que são necessários para apoiar a tarefa atual.

Desta forma, a definição de contexto e o discernimento do comportamento desejável por parte do sistema, em reconhecendo ao contexto corrente, é tarefa inicial para desenvolvimento de sistemas sensíveis ao contexto (context-aware). Sistemas sensíveis ao contexto são definidos como sistemas que utilizam contexto para prover informações e/ou serviços relevantes ao usuário, onde a relevância depende das tarefas desempenhadas pelo usuário (DEY, 2001). Além de serem dependentes do contexto do usuário, sistemas sensíveis ao contexto precisam necessariamente estar aptos a entender e a reagir de forma ativa aos eventos que ocorrem ao seu redor. Eventos dizem respeito a todas as ocorrências de modificação relevantes ao contexto corrente, podendo aqui o termo 'contexto' se referir a dados geográficos, comportamentais, entre outros.

Não se pode tratar qualquer aplicação que considera informações relevantes do usuário como sensível ao contexto, mas apenas aquelas que são principalmente guiadas pelo contexto do usuário.

Segundo Fleischmann (2012), com relação aos dados que são pertinentes para definição do contexto de execução e do usuário, pode-se dividir o contexto em dois grupos: os de dimensões externas, onde os dados buscados são relativos às informações do ambiente em

si, como informações de sensores e de dimensões internas, em que as informações relativas ao usuário são mais importantes, como tarefas, objetivos e atividades executadas.

Atualmente, uma denominação que tem sido muito utilizada é a de aplicações *context-aware*, onde situações são vistas como contextos logicamente interrelacionados. Esta forma de tratar a adaptação ao contexto tem sido adotada em vários trabalhos, pois através dela objetiva-se pré-determinar possíveis situações pelas quais o usuário deve passar no ambiente.

Segundo Fleischmann (2012), quanto melhor o tratamento dado ao contexto, mais complexas são as atividades permitidas dentro do ambiente. Aplicações iniciais eram simples, lidavam apenas com mudanças de localização e de temperatura. Agora, por exemplo, as aplicações já procuram se adaptar às intenções do usuário, buscando modelar todo o seu cenário de atuação. De acordo com as diferentes abordagens dadas para contexto nos trabalhos analisados, a Figura 12, apresentada por Fleischmann (2012), sumariza as abordagens apresentadas, tendo como ponto de partida a divisão existente entre dimensões de contexto externas e internas.

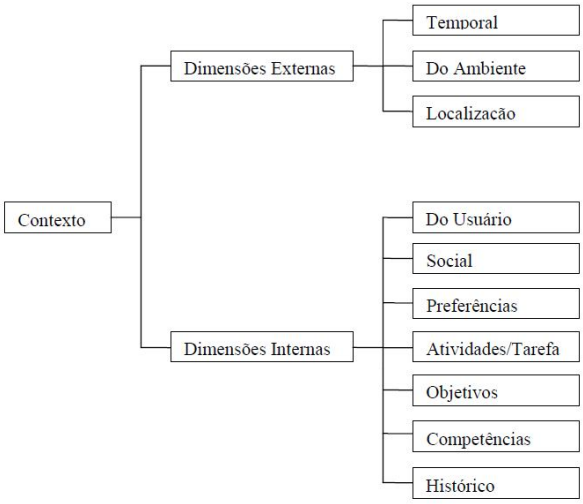


Figura 12: Sugestão de classificação dos tipos de contexto (FLEISCHMANN, 2012)

O contexto de dimensão externa, segundo Fleischmann (2012), pode ser dividido da seguinte forma:

- Temporal - consiste de informações relativas ao tempo ou instante em que determinada instância de contexto ocorre. Pode ser tratado em função de ocorrido no contexto do poder ser independente da início/fim de determinada situação ou evento ambiente. É definido como contexto externo por ação do usuário. A sua granularidade é variável.
- Do Ambiente - consiste de elementos específicos do ambiente em questão,

como temperatura, espaço físico ocupado (tamanho), umidade, condições de operação.

- Localização - se refere a elementos de contexto específicos de localização. A localização é definida como contexto externo pelo mesmo motivo do contexto temporal, no qual a localização não se refere necessariamente ao usuário, mas a qualquer recurso relevante ao sistema.

O contexto de dimensão interna, segundo Fleischmann (2012), pode ser definido da seguinte forma:

- Do Usuário - se refere a elementos de contexto que identificam o usuário e que não variam continuamente no tempo.
- Social - dita o contexto social do usuário no ambiente. Os contextos sociais variam de acordo com a aplicação em questão, por exemplo, em um ambiente de aprendizagem o contexto social pode definir se um usuário é professor ou aluno, ou ainda sua relação com os demais usuários.
- Preferências - define quais as preferências do usuário, isto é, dependendo da aplicação em questão, quais itens são preferidos pelo usuário, em detrimento de outros. Seguindo o exemplo de aprendizagem, as preferências podem determinar quais tipos de OAs são preferidos pelo aluno durante o seu aprendizado. Por exemplo, o aluno pode preferir assistir a um vídeo explicativo ao invés de ler algum texto ou manual para aprender determinado tópico.
- Atividades ou Tarefas - definem as atividades ou tarefas que um usuário necessita seguir para atingir determinado objetivo. Pode ser uma sequência de cursos que um aluno deve fazer para finalizar um módulo de aprendizagem, ou a sequência de tarefas que um professor precisa atribuir aos alunos até o final do semestre.
- Objetivos - são os objetivos finais do usuário de acordo com determinado conjunto de tarefas definidas. Um usuário pode ter mais de um objetivo em um mesmo ambiente, dependendo dos outros contextos em que está inserido.
- Competências - ou conhecimentos, significam as aptidões do usuário para determinada área ou atividade.
- Histórico de Acessos - possui natureza um pouco distinta dos demais tipos de contexto. Descreve as últimas ações desempenhadas pelo usuário no ambiente. Pode ser usado em conjunto com outras dimensões de contexto como, por exemplo, tempo e localização, auxiliando na definição de novos contextos.

A localização do aluno, como apresentado, é uma característica importante para a definição do seu contexto em um ambiente para aprendizagem. Entretanto, o contexto inclui mais do que apenas a localização (MOORE et al. 2009). De fato, quase todas as informações disponíveis no momento da interação podem ser vistas como informações contextuais, dentre as quais destacam-se:

- a variada gama de softwares e tipos de mídias de OA;
- disponibilidade de recursos (hardware e arquitetura utilizada);
- idade;
- histório escolar (reprovação e área de conhecimento);
- informação temporal (ex. hora do dia, data, entre outros);
- necessidades especiais (definciências auditivas, visuais, entre outras);
- conhecimento em linguas estranjeiras.

Esta lista apresentada serve para demonstrar a complexidade do contexto de um aluno, sua natureza específica e a dificuldade em defini-la e medi-la.

Na tentativa de facilitar a compreensão sobre o contexto, (MOORE et al. 2008) definem uma outra forma de classificação de contexto: estático (denominado customização) e dinâmico (denominado personalização). O primeiro diz respeito à situação na qual um perfil de usuário é criado manualmente, estando o usuário ativamente envolvido no processo e tendo um elemento de controle. Já o contexto dinâmico refere-se à situação na qual o usuário é visto como sendo passivo, ou pelo menos com um pouco menos de controle no processo de criação de seu perfil. Nesse caso, o sistema monitora, analisa e reage dinamicamente ao comportamento do usuário e ao perfil identificado.

3.2.2 AMBIENTES SENSÍVEIS AO CONTEXTO

Um sistema é considerado sensível ao contexto (*context-aware environments*) se ele utiliza o contexto de uso para fornecer informações ou serviços relevantes para o usuário (DEY, 2001).

Segundo Vieira et al. (2009):

> "Sistemas Sensíveis ao Contexto (SSC) são aqueles que gerenciam elementos contextuais relacionados a uma aplicação em um domínio e usam esses elementos para apoiar um agente na execução de alguma tarefa".

São tarefas dos SSCs, por exemplo, apoiar os alunos a atingirem uma meta específica de aprendizagem, fornecendo a eles anotações contextualizadas ou recomendando uma

sequência de recursos de aprendizagem (HINZ et al. 2011).

Já Rodolpho (2009) destaca que a fim de desenvolver ambientes de aprendizagem que sejam sensíveis ao contexto do estudante, é fundamental que os conteúdos educacionais sejam criados de maneira padronizada. Desta forma, é possível que um ambiente de suporte à aprendizagem exiba o conteúdo de forma adequada e reutilize conteúdos em diferentes contextos e a partir de diversos repositórios. Além disso, é possível combinar conteúdos distintos, o que, por sua vez, melhora o processo de produção de conteúdo e, como conseqüência, reduz os seus custos (RODOLPHO, 2009). Um modo eficaz de padronizar conteúdos educacionais é através do uso de Objetos de Aprendizagem.

Segundo Fleischmann (2012), a adaptação e sensibilidade ao contexto são temas que andam juntos em diferentes momentos. A sensibilidade ao contexto não necessariamente implica adaptação, mas um sistema adaptativo, em geral, precisa estar ciente da ocorrência de determinados eventos no sistema e reagir de forma adequada a estes eventos. Independentemente da aplicação, pode-se citar uma série de áreas que têm feito uso de adaptação e/ou sensibilidade ao contexto, sendo mais característico de aplicações autodenominadas móveis, pervasivas e/ou ubíquas.

Fleischmann (2012) complementa que nestes tipos de sistemas a sensibilidade ao contexto é usada para manter o sistema consistente em termos de funcionamento, e também mantê-lo capaz de proativamente perceber as melhores condições de ação e agir de acordo com esta percepção. Entretanto, a tarefa de adaptação é bastante complexa, uma vez que para determinar as alterações no contexto do ambiente muitas vezes não se avalia somente modificações isoladas, mas de vários aspectos combinados.

A mobilidade e facilidade de acesso aos sistemas *e-learning* por meio de novas tecnologias, como *Tablets*, *SmartPhones* e PDAs (*Personal Digital Assistants*), tem também impulsionado estes avanços, onde o problema da mobilidade não necessariamente diz respeito ao dispositivo computacional utilizado, mas ao próprio usuário, que está em constante movimento e acessa ao ambiente *e-learning* em diferentes locais através de diversos tipos de dispositivos computacionais, móveis ou não. O desafio, neste caso, é personalizar a aplicação Web independentemente do contexto atual do usuário.

Algumas aplicações de aprendizagem sensíveis ao contexto utilizam contextos de aprendizagem a fim de adaptar ou sugerir apropriadamente atividades e conteúdos.

3.3 REDE DE ONTOLOGIAS

O problema das diferenças sintáticas e semânticas entre os dados de diferentes modelos de

aluno pode ser superado com o uso de ontologias. Gruber (1993) descreve uma ontologia como “uma especificação explícita de uma conceitualização”, ou seja, uma formalização dos conceitos e relacionamentos em um domínio.

Já Yu (2011) analisa ontologia em três conceitos (domínio, classes e propriedades). O primeiro conceito de ontologia diz respeito ao domínio modelado. Ele é utilizado para descrever uma área específica do conhecimento que será abordada na modelagem da ontologia, como, por exemplo, medicina, finanças e fotografia. O segundo conceito importante é o de termos. Os termos são usualmente chamados de classes e a relação entre termos pode ser expressa através de uma estrutura hierárquica, possuindo classes superiores (superclasses) que representam os termos mais abrangentes e classes inferiores (subclasses) que representam termos mais detalhados, sendo que as subclasses possuem os mesmos atributos e características de suas superclasses (conceito de herança). O último conceito importante é outra relação entre termos utilizando um grupo especial de termos chamados de propriedades. As propriedades descrevem vários atributos e características das classes, podendo também associar diferentes classes. Em verdade, também é possível construir relações hierárquicas entre termos distintos que deem origem a um subtermo (subclasse) comum através das propriedades.

Yu (2011) também acrescenta os benefícios de se utilizar ontologias:

- possibilita um entendimento comum e compartilhado sobre conceitos em um domínio específico.
- permite a reutilização do domínio de conhecimento.
- juntamente com as linguagens baseadas em lógica descritivas, como, por exemplo, a OWL (*Ontology Web Language)* e o RDF (*Resource Description Framework*) que permite codificar o conhecimento e a semântica para que máquinas possam compreender.
- torna possível que o processamento do domínio por máquinas.

A ontologia representa as informações de todo um domínio, o que permite que sua representação possa ser utilizada por diversas pessoas, base de dados e aplicações do domínio. Assim, outras aplicações que necessitam destas informações podem utilizar essa representação, ao invés de ter que defini-las novamente.

Segundo Fleischmann (2012) são usadas tanto para modelagem como para raciocínio sobre o contexto e, em geral, para a descoberta de conhecimento, são utilizadas com apoio de ferramentas de *reasoning*. De acordo com D'Aquin, Gangemi e Haase (2006) apud

Fleischmann (2012), ontologias na Web não podem ser tratadas como artefatos isolados, pois estão relacionadas umas às outras de uma forma que pode afetar seu significado por serem inerentemente distribuídas em uma rede de recursos semânticos interligados. Desta forma, os autores definem uma rede de ontologias como uma coleção de ontologias relacionadas umas com as outras a partir de uma variedade de tipos de relacionamentos.

Segundo Fleischmann (2012), uma rede de ontologias difere de um conjunto de ontologias interconectadas devido à meta-relacionamento definido entre as ontologias envolvidas. Existem diversos tipos de meta-relacionamentos:

- dependências e importações: consistem do tipo mais simples de relacionamento, ocorrendo quando uma ontologia, para definir seu próprio modelo, requer a referência a definições incluídas em outra ontologia.
- versionamento: consiste da atividade de manter o registro das diferentes versões existentes de uma mesma ontologia.
- alinhamento: se refere à técnica de colocar planos diferentes em correspondência, declarando quais de suas entidades devem ser consideradas como sendo as mesmas ou como sendo mais genéricas do que as demais.
- modularização: uma ontologia é a modularização de outra quando consiste de uma divisão do modelo ontológico como um todo, sendo autocontida e mantendo seus componentes interligados, os quais podem ser considerados independentemente, mas que participam cada um em um aspecto específico ou subdomínio da ontologia.

3.4 PADRÕES E MODELOS DE DADOS DE ALUNOS

Para que um sistema computacional possa recomendar objetos de aprendizagem a um aluno é necessário que o mesmo conheça suas características, ou seja, o sistema deve obter e gerenciar algumas informações básicas sobre o perfil e contexto do aluno, saber suas preferências, seus objetivos e estilos de aprendizagem. Além disso, estas informações devem estar padronizadas a fim de evitar inconsistências semânticas e facilitar a interoperabilidade entre diversos módulos e sistemas (HINZ et al. 2011).

A fim de estabelecer um padrão que seja essencial ao intercâmbio destas informações, as organizações internacionais como IEEE (*Institute of Electrical and Electronic Engineers*), IMS (*Instructional Management Systems*) e ADL (*Advanced Distributed Learning*) trabalham na padronização de informações de ambientes de aprendizagem, especificando metamodelos

de dados para informações sobre os alunos (HINZ et al. 2011).

Dentre as principais especificações podem ser citadas: o EduPerson (EDUPERSON, 2005), o ULF (*Universal Learning Format*) (SABA, 2000), o PAPI (*Personal and Private Information*), da IEEE (PAPI, 2001), o IMS LIP (*Instructional Management Systems Learning Information Package*), da *Global Learning Consortium* (IMS LIP, 2007).

3.4.1 EDUPERSON

EduPerson (EDUPERSON, 2005) foi desenvolvido com o objetivo de facilitar a comunicação entre os sistemas das instituições de ensino superior. A missão desse grupo era definir uma classe de objetos para descrever pessoas em universidades para ser utilizada em diretórios LDAP (*Lightweight Directory Access Protocol*). O grupo de trabalho também definiu recomendações sobre a semântica, sintaxe e a privacidade dos dados.

A versão 1.6 do EduPerson (Tabela 4) possui 42 atributos divididos em duas categorias: os atributos gerais (atributos comuns como nome, e-mail, endereço, entre outros) e os atributos novos, que foram criados para facilitar a colaboração interinstitucional das aplicações. Cada atributo EduPerson (EDUPERSON, 2005) tem uma classe utilitária de aplicação associada, a qual sugere a classe de aplicações para as quais esse atributo é apropriado.

Tabela 4: Atributos do EduPerson (EDUPERSON, 2005)

Categoria	Atributo	Descrição
Gerais	*Common Name*	Atributo obrigatório que contém o nome completo
	Description	Contém uma descrição sobre o comportamento da pessoa. Por exemplo: otimista
	Display Name	O nome que deve ser usado
	Facsimile Telephone Number	Número de fax
	Home Phone	Número de telefone residencial
	Jpeg Photo	Usado para armazenar imagens da pessoa
	Locality Name	Nome de uma localidade (cidade, país)
	Mail	Endereço eletrônico
	Preferred Language	Língua preferida da pessoa
	Surname	Atributo obrigatório que contém o sobrenome da pessoa
	User Password	Identifica a senha
Novos	*EduPerson Affiliation*	Relação da pessoa com a instituição, como: aluno, entre outros
	EduPerson Entitlement	Relação com a função ou cargo
	EduPerson Nickname	Nome da pessoa (Apelido)
	EduPerson Org DN	Instituição com a qual a pessoa tem ligação
	EduPerson Org Unit DN	Unidades organizacionais
	EduPerson Primary Affiliation	Relação principal da pessoa com a instituição
	EduPerson Primary Org Unit DN	Principal unidade organizacional
	EduPerson Primary Name	Nome principal para autenticação inter institucional

A utilização do EduPerson (EDUPERSON, 2005) não implica no uso de todos os seus

atributos, os únicos atributos obrigatórios são *common name* e *surname* (nome e sobrenome), quanto aos outros atributos, à instituição é que decide se irá utilizá-los ou não.

3.4.2 UNIVERSAL LEARNING FORMAT

O ULF (*Universal Learning Format)* (SABA, 2000) é um conjunto de formatos baseados em XML (*Extensible Markup Language*). O ULF possui o objetivo de fornecer portabilidade para dados educacionais, sendo que os formatos que os compõe podem ser mapeados para outros padrões.

O ULF é composto por cinco elementos: *Catalog Format* (Formato Catálogo), *Learning Content Format* (Formato de Conteúdo Educacional), *Competency Format* (Formato de Competência), *Certification Format* (Formato de Certificação) e *Profile Format* (Formato do Perfil).

O elemento *Profile Format* (Formato do Perfil) descreve os dados do perfil do aluno e compreende vários tipos de informações, como por exemplo, dados pessoais, objetivos, histórico educacional, competências adquiridas e certificações. Ele é uma representação em XML para escrever informações de perfil de aluno. Essas informações são capturadas em XML e utilizam RDF para definir metadados que descrevam o aluno. Ele também incorpora padrões de metadados existentes como o *vCard* (VCARD, 1996), os quais garantem compatibilidade com descrições de perfil já desenvolvidas (SABA, 2000) (Tabela 5).

Um documento *Profile Format* (Formato do perfil) contém pelo menos um elemento *Description* (Descrição), que descreve um aluno no sistema e possui um conjunto de propriedades, bem como um identificador único. O atributo identificador pode identificar o aluno ou a URL de um recurso que é utilizado quando a informação do aluno é definida em outro local, como uma página web.

Tabela 5: Atributos do ULF (SABA, 2000)

Categoria	Atributos	Descrição
Personal Information (Informações pessoais)	*Name*	Nome
	Birthdate	Data de nascimento
	Electronic mail	Endereço eletrônico
	Organization Name	Nome da organização
	Unit Identifier	Identificador da unidade
Learning Information (Informações de aprendizagem)	*Status*	Status
	Result	Resultado
	Date	Data
	Score	Nota
Goal Information (Informações do objetivo)	*Description*	Descrição
	Start Date	Data de início
	End Date	Data final

	Status	Status
Observation Information (Informações de observação)	*Comment*	Comentário
	Goal	Objetivo
	Competency	Competência
	Details	Detalhes
Competency Information (Informações de competência)	*Competency*	Competência
	Details	Detalhes
Certification Information (Informações de certificação)	*Certification*	Certificação
	Status	Status
Preference Information (Informações de preferências)	*Language*	Idioma
	Country	País
Profile Information (Informações de perfil)	*Description*	Descrição
	Publisher	Editor
	Language	Idioma

3.4.3 IEEE PAPI *(PUBLIC AND PRIVATE INFORMATION LEARNER)*

A especificação PAPI *(Public and Private Information Learner*) (PAPI, 2001) originou-se em 1998 no grupo LTSC *(IEEE Learning Technology Standards Committee*). Porém, em 2001, o seu desenvolvimento foi transferido para o *Working Group Subcommittee* que desenvolve padrões internacionais de dados de aluno para a ISO/IEC (*Internacional Organization for Standardization / International Electrotechnical Commission*).

O PAPI especifica a sintaxe dos dados do aluno caracterizando suas informações pessoais, seus conhecimentos, preferências, entre outras (PAPI, 2001). A sua principal característica é a divisão das informações em seis categorias principais, que visam organizar os diversos tipos de dados do aluno. A Figura 13 apresenta as categorias do PAPI.

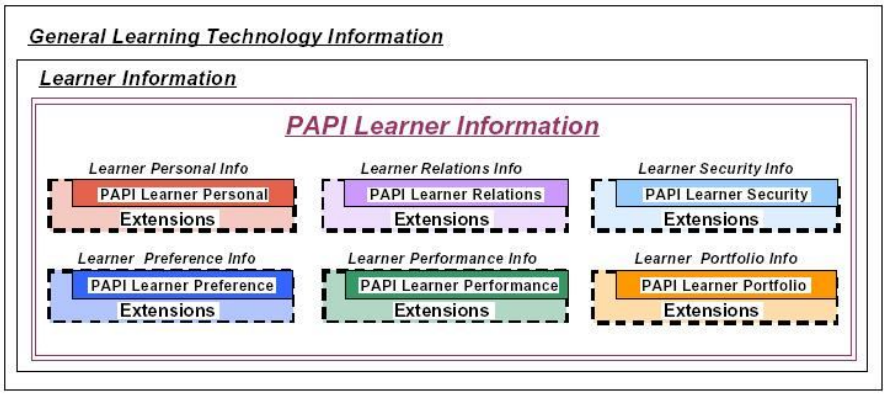


Figura 13: Categorias PAPI (PAPI, 2001)

A Categoria *Learner Personal* (Dados Pessoais) refere-se aos dados pessoais do aluno, como nome, endereço, telefone, e-mail, entre outros. O *Learner Relations* (Relacionamentos do Aluno) reporta os relacionamentos do aluno com outros usuários do sistema de EaD, como

professores, entre outros. A categoria *Learner Security* (Segurança do Aluno) preocupa-se com as credenciais de segurança como: senhas, chaves de criptografia públicas e privadas. As preferências do aluno fazem parte da categoria *Preference* (Preferências). O desempenho do aluno em cursos já realizados, bem como o seu histórico é descrito na categoria *Performance* (Desempenho). *Portfólio* (Portfólio) possui as experiências anteriores do aluno e descreve um conjunto de trabalhos já realizados que visam ilustrar e justificar as habilidades do estudante.

Não é definida no padrão a forma de combinar essas categorias, o que permite que sistemas que o adotarem agrupem os dados da forma que acharem mais adequados. O PAPI (PAPI, 2001) não especifica todas as informações possíveis sobre o aluno, mas permite que sistemas definam suas próprias extensões ou utilizem apenas as categorias que acharem mais relevantes para a aplicação.

Alguns aspectos quanto à segurança dos dados são discutidos no documento WG3/N0017 (PAPI, 2001) do PAPI. Esse documento contém questões gerais de segurança que deverão ser consideradas na implementação do PAPI, as quais são apenas recomendações e não normas que devem ser seguidas.

Privacidade e integridade dos dados são critérios que não são considerados no PAPI e são de extrema importância, pois geralmente em um modelo de aluno existem dados pessoais. Esses tipos de dados, por exemplo, são informações privadas, diferentemente de dados contidos no portfólio do aluno, que são de acesso público. A natureza da informação (privada, pública, entre outros), bem como os mecanismos e técnicas que devem ser considerados para implementação estão fora do escopo do PAPI (Tabela 6).

Tabela 6: Atributos do PAPI (PAPI, 2001)

Categoria	**Atributos**	**Descrição**
Personal (Dados Pessoais)	*My_contact_identifier_list*	Identificador do contato
	Name_list	Nome do aluno
	Telephone_list	Telefones do aluno
	Email_contact_list	E-mail do aluno
	Postal_address_list	Endereço postal do aluno
	Contact_bucket	Extensão para informações de contato
	Us_social_security_number	Documento
Relations (Relacionamentos)	*Others_identifier_list*	Pessoas que possuem algum tipo de relacionamento com o aluno
	Relations_label_list	Descreve a natureza do relacionamento
	Relation_to_them_list	Como está relacionado com os outros
	Relation_to_me_list	Como os outros estão relacionados com o aluno
Security (Segurança)	*My_security_identifier_list*	Identificador de segurança
	Credential_list	Lista das credenciais de segurança
	Security_bucket	Extensão de elementos de segurança
Preference (Preferências)	*My_preference_identifier*	Identificador de preferência
	Preference_hid_list	Identificador externo para relacionar repositórios

	Preference_name	Grupo de preferência
	Pre_include_preference_list	Preferências incluídas antes da lista
	Post_include_preference_list	Preferências incluídas depois da lista
	Hci_device_preference_list	Lista de preferências de dispositivos para interface homem-computador (texto, gráfico, áudio, vídeo, entre outros)
	Preference_bucket	Extensão de preferências
Performance (Desempenho)	*My_performance_identifier_list*	Identificador de desempenho
	Performance_hid_list	Identificador externo para relacionar repositórios
	Owner_identifier	Identificador do responsável
	Recording_date_time	Data-hora do registro
	Valid_date_time_begin	Data e hora de início
	Valid_date_time_end	Data e hora de término
	Issue_from_identifier	Pessoa responsável pela avaliação
	Issue_date_time	Data e hora da avaliação
	Issue_to_identifier	Entidade responsável
	Learning_experience_identifier	Identificador da experiência educacional
	Competency_identifier	Identificador da competência
	Granularity	Granularidade
	Performance_coding_scheme	Tipo de codificação, medida utilizada
	Performance_metric	Intervalo de valores permitidos
	Performance_value	Valor da avaliação
	Certificate_list	Lista de certificados
	Performance_bucket	Reservado para extensão
Portfolio (Portfólio)	*My_portfolio_identifier_list*	Identificadores do portfólio
	Portfolio_hid_list	Identificador externo para relacionar repositórios
	Media_id_kind	Tipo de mídia
	Media_id	Identificador da mídia
	Media_lom_list	Lista de objetos
	Media_papi_learner_performance_list	Lista de referências do desempenho do aluno
	Media_competency_definition_list	Lista de mídias de competências
	Certificate_list	Lista de certificação
	Portfolio_bucket	Campo para extensão

3.4.4 *IMS LEARNER INFORMATION PACKAGE*

A especificação LIP (*Learner Information Package*) (IMS LIP, 2007) foi definida pelo *Global Learning Consortium* da IMS (*Instructional Management Systems*), que é um consórcio de instituições educacionais, companhias de software e editoras.

O objetivo do consórcio é promover a disseminação de especificações que permitam ambientes de aprendizagem distribuídos (IMS LIP, 2007). De 1997 a 1999, a IMS trabalhou em conjunto com o grupo LTSC na definição do PAPI (PAPI, 2001), porém no final de 1999, a IMS decidiu produzir sua própria especificação para dados de alunos. Finalmente, em 2001 foi produzido o documento LIP que inclui o modelo de dados e esquemas XML para implementação do modelo.

No LIP as informações do aluno são separadas em onze categorias, como mostra a Figura 14.

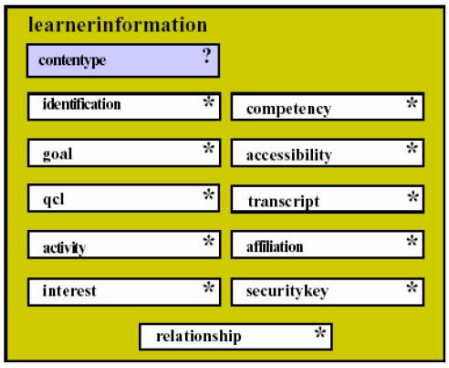


Figura 14: Categorias do LIP

A Categoria *Identification* (Identificação) contém os dados demográficos e biográficos do aluno. A categoria *Goal* (Objetivos) descreve os objetivos do aluno. Em QCL (Qualificações, Certificados e Licenças) são descritas suas qualificações, diplomas e certificados. Qualquer atividade relacionada com estudo, já concluído ou em andamento, está referenciada na categoria *Activity* (Atividades).

Os interesses do aluno e seus hobbies estão descritos na categoria *Interest* (Interesse). A categoria *Relationship* (Relacionamentos) contém os relacionamentos definidos entre o aluno e sua identificação, acessibilidade, qualificações, competências, objetivos, atividades, interesses, chaves da segurança e afiliações.

As habilidades do aluno, experiências e conhecimento já adquiridos estão na categoria *Competency* (Competências). Dados sobre a acessibilidade da informação ao aluno, idioma e preferências (físicas e tecnológicas), fazem parte da categoria *Accessibility* (Acessibilidade), *Transcript* (Transcrição) contém um resumo por instituições das realizações acadêmicas do aluno. *Affiliation* (Afiliação) possui informação sobre membros de organizações profissionais. *Security key* (Chave de segurança): informação de segurança relacionada ao aluno.

Em cada categoria foram definidos elementos, que são na maioria opcionais, e estruturas de dados que podem ser estendidas (Tabela 7). Em cada categoria LIP são associados os metadados: *referencial* (referencial), *temporal* (temporal) e *privacy* (privado). O metadado *referencial* (referencial) define a estrutura que contém o identificador único do dado. O metadado *temporal* (temporal) contém informações de tempo relacionadas com o dado, como, por exemplo, data da criação, validade, entre outros. O metadado *privacy* (privado) contém os dados relevantes para privacidade. Não é especificado que tipo de dado ou padrão deve ser usado nos metadados *referencial* (referencial) e *privacy* (privado).

Um dos maiores problemas do LIP é apresentar categorias conflitantes e que se sobrepõem, como por exemplo, as categorias QCL (Qualificações, Certificações e Licenças),

activity (atividade), *transcript* (transcrição) e *competency* (competência).

Quanto à privacidade dos dados, LIP permite associar uma descrição do tipo de privacidade que deve ser usada em cada categoria. Mecanismos que devem ser implementados ou arquiteturas que possam ser empregadas para suportar a privacidade dos dados não são explicitados no LIP. Quanto à segurança de acesso aos dados, LIP apresenta a estrutura de dados *security-key* (chave de segurança), que incluiu senhas, chaves públicas e assinatura digital.

Em LIP é descrito um esquema XML onde são definidos os elementos de dados, conteúdo, atributos e um arquivo DTD (*Document Type Definition*), que estabelece os elementos e conteúdo que podem ser usados em cada categoria.

Tabela 7: Atributos do LIP (IMS LIP, 2007)

Categoria	Atributos	Descrição
Identification (Identificação)	*Comment*	Comentário sobre a informação
	Contentype	Tipo de conteúdo
	Formname	Formato do nome da pessoa
	Name	Nome da pessoa
	Address	Endereço da pessoa
	Contactinfo	Informação de contato da pessoa
	Demographics	Demografia
	Identification	Data de nascimento, foto, sexo e local
	Agent	Dados sobre os agentes que podem agir em benefício do aluno (parente, tutor, entre outros)
	Extension	Reservado para extensão
Acessibility (Acessibilidade)	*Comment*	Comentários sobre a categoria (idiomas de leitura, escrita e fala do aluno)
	Contentype	Dado que é usado para descrever o conteúdo da estrutura
	Language	Idioma usado
	Preference	Preferências de ensino
	Extension	Reservado para extensão
Goal (Objetivos)	*Typename*	Tipo de objetivo do aluno
	Comment	Comentários da informação
	Contentype	Dado usado para descrever o conteúdo
	Date	Datas registradas para o objetivo
	Priority	Prioridade do objetivo
	Status	Status registrado do objetivo
	Description	Descrição do objetivo
	Goal	Sub objetivos
	Extension	Campo para extensão
QCL (Qualificações, Certificados e Licenças)	*Typename*	Tipo de qualificações, certificações e licenças
	Comment	Comentários sobre a informação
	Contentype	Dado usado para descrever o conteúdo
	Title	Título da qualificação, certificado ou licença
	Organisation	Organização responsável pela qualificação
	Registration	Número de registro
	Level	Nível da qualificação
	Date	Data registrada
	Description	Descrição da qualificação, licença ou certificado

	Extension	Campo para extensão
Activity (Atividades)	*Typename*	Tipo de atividade ou treinamento
	Service comment	Comentários sobre a informação
	Contentype	Dado usado para descrever o conteúdo
	Date	Data de registro da atividade
	Status	Status da atividade
	Units	Unidade da atividade
	Learningactivityref	Referência externa para um identificador de ensino associado à atividade
	Definition	Definição da atividade
	Product	Produto criado como parte obrigatória da tarefa
	Testimonial	Recomendação para o aluno dado por alguém associado com a atividade
	Evaluation	Avaliação da atividade
	Description	Descrição da atividade
	Activity	Sub atividade
	Extension	Campo para extensão
Competency (Competências)	*Comment*	Comentários sobre a informação
	Contentype	Dado usado para descrever o conteúdo
	Exrefrecord	Identificador externo para relacionar repositórios
	Description	Descrição da competência
	Extension	Campo para extensão
Interest (Interesses)	*Typename*	Tipo de interesse
	Comment	Comentários sobre a informação
	Contentype	Dado usado para descrever o conteúdo
	Product	Produto criado como parte de uma atividade de interesse
	Extension	Campo para extensão
Affiliation (Afiliação)	*Typename*	Tipo de afiliação
	Comment	Comentários sobre a informação
	Contentype	Dado usado para descrever o conteúdo
	Classification	Tipo de afiliação no grupo, membro, amigo, entre outros
	Affiliationid	Número do grupo
	Role	Papel assumido no grupo
	Organisation	Organização onde é afiliado
	Date	Data do registro
	Status	Status da afiliação
	Description	Descrição da afiliação
	Affiliation	Referência recursiva
	Extension	Campo para extensão
Transcript (Transcrição)	*Typename*	Tipo de transcrição
	Comment	Comentários sobre a informação
	Contentype	Dado usado para descrever o conteúdo
	Exrefrecord	Identificador externo para relacionar repositórios
	Description	Descrição da transcrição
	Extension	Campo para extensão
Security key (Chave de Segurança)	*Typename*	Tipo de chave de segurança
	Comment	Comentários sobre a informação
	Contentype	Dado usado para descrever o conteúdo
	Keyfields	Classificação da chave, por exemplo: senha
	Description	Descrição da chave
	Extension	Campo para extensão
Relationship (Relacionamentos)	*Typename*	Tipo de relacionamento
	Comment	Comentários sobre a informação
	Contentype	Dado usado para descrever o conteúdo

	Tuple	Tupla que define o relacionamento 1 pra n
	Description	Descrição do relacionamento
	Extension	Campo de extensão

3.4.5 ANÁLISE COMPARTIVA DOS PADRÕES E MODELOS

Foi realizado um estudo comparativo entre as categorias, que visava demonstrar a diversidade de elementos e a abrangência de cada padrão. É através desse estudo que foi possível identificar que a comunidade de pesquisa dessa área, reconhece a importância de definir um modelo dados de aluno padrão, que possa ser compartilhado entre diversos sistemas.

Além disso, verifica-se que um importante requisito ainda não é totalmente atendido: a interoperabilidade desses dados. Os padrões definem o conjunto de dados necessário, porém o domínio de cada dado não é explicitado. Os padrões definiram a maioria dos campos como do tipo texto, que permite inserir texto livre. Isso é um dos grandes problemas para a interoperabilidade, pois a falta de padronização nas instâncias gera diversos problemas, pois, por exemplo, um registro sobre o desempenho do aluno deve ter uma classificação padrão, pois um sistema pode representar notas sobre uma atividade usando uma classificação de 1 a 10, enquanto que outro sistema pode utilizar conceitos como A, B ou C.

A Tabela 8 apresenta a relação entre as categorias PAPI (PAPI, 2001), LIP (IMS LIP, 2007) e ULF.

Tabela 8: Relação entre as categorias dos modelos

Categorias IEEE PAPI	**Categorias IMS LIP**	**Categorias ULF**
Personal Information	*Identification*	*Personal*
Relations	*Relationships /affiliation*	*-*
Preference	*Accessibility, Interest*	*Preference*
Performance	*Transcript, goal, qcl, activity*	*Learning / Observation*
Portfolio	*Activity, competency*	*Competency /Certification/ Goal*
Security	*Security key*	*-*
-	*-*	*Profile Information*

As categorias *Relations* (Relacionamentos) (PAPI) e *Affiliation* (Afiliação) (LIP) descrevem os relacionamentos que o aluno possui, sendo essa informação não contemplada no ULF.

A categoria *Affiliation* (Afiliação) permite o registro de organizações pelas quais o aluno teve algum tipo de relacionamento (aluno, funcionário, entre outros), bem como o período desse relacionamento. Essa categoria entra em conflito com outras categorias do próprio LIP como: *Activity* (Atividade), *Competency* (Competências) e QCL (Qualificações, Certificados e Licenças), que também podem possuir esse tipo de informação. LIP também apresenta a categoria *Relationship* (Relacionamento), que serve apenas para mapear

relacionamentos entre os tipos de categorias do LIP, como, por exemplo, uma instância de *Relationship* (Relacionamento) pode informar que um registro da categoria QCL tem relação com um determinado registro da categoria *Transcript* (Transcrição).

A categoria *Preference* (Preferências) do PAPI permite descrever qualquer tipo e informação relacionada com as preferências do aluno, tanto no nível de interesse pessoal quanto no nível de acessibilidade. Essas informações estão divididas em duas categorias no LIP: *Accessibility* (Acessibilidade) e *Interest* (Interesse). *Accessibility* (Acessibilidade) consiste apenas das preferências do aluno quanto a idiomas e questões técnicas. A categoria *Interest* (Interesses) do LIP contém as informações de interesse pessoal do aluno como seus hobbies, entre outros. ULF possui a categoria *Preference Information* (Informação de Preferência) que permite qualquer tipo de informação relacionada com preferências do aluno. Uma das vantagens que o PAPI apresenta nesse tipo de categoria é a possibilidade de estabelecer prioridades entre as preferências.

Informações sobre qualificações, certificados, licenças, atividades realizadas, objetivos e competências estão referenciadas na categoria *Performance* (Desempenho) do PAPI. No LIP esse tipo de informação está separado por categoria, ou seja, existe uma categoria para cada tipo de informação: QCL (Qualificações, Certificados e Licenças), *Activity* (Atividade), *Competency* (Competências), *Goal* (Objetivos), *Transcript* (Transcrição). O mesmo ocorre no modelo ULF, esse tipo de informação está dividido entre as categorias *Certification Information* (Informação de certificado), *Competency Information* (Informação de competência) e *Goal Information* (Informação de objetivo).

Os critérios: segurança, privacidade, extensibilidade, flexibilidade e suporte a implementação, também foram considerados para comparação dos modelos. A Tabela 9 apresenta o comparativo realizado entre os modelos PAPI (PAPI, 2001), IMS LIP (IMS LIP, 2007) e ULF.

Quanto à segurança e privacidade dos dados o PAPI define alguns mecanismos para prover segurança e privacidade dos dados. O LIP permite a inclusão de mecanismos para manter a privacidade e proteger a integridade dos dados. O padrão ULF não trata da questão de privacidade e segurança dos dados, que são itens importantes para assegurar um intercâmbio de dados consistente e seguro.

Tabela 9: Comparativo entre os modelos

	PAPI	**IMS LIP**	**ULF**
Segurança	Parcialmente	Parcialmente	Não possui
Privacidade	Parcialmente	Parcialmente	Não possui
Extensibilidade	Plenamente	Plenamente	Parcialmente

Suporte a implementação	Parcialmente	Plenamente	Não possui

Quanto à extensibilidade e flexibilidade, os padrões permitem a inclusão de outros tipos de informação, possibilitando a sua extensão. Quanto ao suporte para o processo de implementação do padrão, o PAPI (PAPI, 2001) apresenta um documento que explicita o mapeamento do modelo conceitual para XML. O LIP apresenta o XML *Binding* que descreve o processo de codificação dos dados e um guia para a implementação do padrão. ULF não apresenta nada em relação a isso.

Em qualquer um dos modelos, o processo de identificação dos alunos é totalmente definido na implementação, sendo que não existe um modelo de confiabilidade, integridade e qualidade dos dados. Nenhum dos modelos define técnicas para validar a informação de desempenho e portfólio do aluno. Além disso, o suporte a múltiplas visões dos dados, protocolos de comunicação e definição de repositórios para armazenamento desses dados não fazem parte do escopo de nenhum dos padrões.

O LIP possui a vantagem de ter sido desenvolvido com base no PAPI (PAPI, 2001). Além disso, o padrão LIP tem maior expressividade na área educacional pelo fato de ser desenvolvido por um consórcio que envolve universidade e empresas.

As características analisadas mostram que os padrões PAPI, LIP e ULF são bastante similares, porém cada um dos padrões apresenta deficiências em algumas características. Há a necessidade de um modelo menor, mais compacto e flexível que possua os elementos mais úteis.

Apesar do objetivo do EduPerson (EDUPERSON, 2005) ser troca de dados de alunos entre instituições de ensino superior, os dados especificados no EduPerson apenas descrevem o aluno em termos gerais, como seus dados pessoais e instituições que ele faz parte. Dados sobre as suas preferências, objetivo, competências e desempenho em atividades, que são necessários, não são abordados. Outro ponto importante a considerar é que o EduPerson foi modelado para um contexto específico, ou seja, voltado para um ambiente de diretórios LDAP, onde a atualização constante desses dados não é de forma facilitada.

O padrão ULF não trata da questão de privacidade e segurança dos dados, que são itens importantes para assegurar um intercâmbio de dados consistente e seguro, pois um modelo de aluno lida com dados privativos e cabe ao aluno escolher se deseja ou não compartilhar seus dados. Além disso, uma vez permitida à troca de dados, essa deve ser realizada de forma segura, com o uso de algum mecanismo como: criptografia, certificados digitais, entre outros.

O PAPI (PAPI, 2001) é uma das principais contribuições para a padronização de dados

de ensino e sua principal vantagem é o fato de ter sido desenvolvido por um grupo de instituições de ensino e empresas bem como, instituições conceituadas como a IEEE e ISO.

Uma limitação do PAPI (PAPI, 2001) é que o mesmo não lida suficientemente com algumas características sobre o aluno como objetivos, preferências e interesses, que são dados essenciais. Um ponto forte do PAPI é o fato dele ter focado a modelagem do aluno em dados sobre o seu desempenho e competências adquiridas ao longo do seu período acadêmico.

Um dos principais problemas do LIP é o fato de ser muito extenso, descrevendo muitas categorias de dados que normalmente não serão utilizadas em um único sistema. O que normalmente acontece é que cada sistema utiliza as categorias que consideram mais relevantes. Além disso, uma limitação de LIP é que ele não apresenta mecanismos técnicos para separação dos dados, isto pode ser um problema quando se combina informações de diferentes fontes ou quando se tem a necessidade de separar diferentes tipos de informações por razões de segurança ou integridade.

3.5 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Apesar da diversidade de definições de contexto, há duas conclusões consensuais relevantes: o contexto atua como um conjunto de restrições que influenciam o comportamento de um sistema embutido em uma dada tarefa e a definição de contexto depende da área de conhecimento (BAZIRE; BRÉZILLON, 2005).

Para técnicas de recomendação deste trabalho, consideramos a avaliação do professor n o plano didático. Nele o professor identificará elementos do contexto do aluno considerando as tarefas e objetivos e competências a serem alcançadas. Considerando a filtragem baseada em conteúdos a arquitetura de recomendação BROAD-RS é sensível ao contexto, adotando dados de perfil e contexto do aluno atuais.

Comparando-se todos estes modelos de dados de alunos (EduPerson (EDUPERSON, 2005), o ULF (SABA, 2000), o PAPI (PAPI, 2001) e o IMS LIP (IMS LIP, 2007)) percebe-se que apesar da diversidade de elementos, muitos metadados são comuns, como por exemplo, informações pessoais (nome, endereço, idade, sexo), as preferências (língua, interface) e o histórico do aluno (tarefas, atividades, leituras).

Ainda não existe um consenso sobre quais dados devem realmente conter um modelo de aluno, bem como não existe um modelo completamente definido ou totalmente implementado, assim, neste trabalho propomos o desenvolvimento de um modelo conceitual mais amplo, que forneça os principais dados sobre o aluno e que possa ser utilizado em sistemas de recomendação, para tal a rede de ontologias garante a semântica da solução.

As informações de contexto usados na arquitetura BROAD-RS, baseadas na abordagem apresentada por Fleischmann (2012), são: dimensões externas (ambiente e localização) e dimensões internas (usuário, social, preferências, atividades, objetivos e competências).

Assim, para suprir as questões acima, foi proposta a ontologia PERSONNA, que tem por objetivo servir como base para modelagem do perfil e contexto do aluno, a fim de facilitar o processo de recomendação de objetos de aprendizagem da arquitetura BROAD-RS, ou seja, filtrar informações relevantes para o aluno e, a partir destas, selecionar objetos de aprendizagem mais apropriados de acordo com seu contexto. A rede de ontologias complementa aontologia PERSONNA ao dipsor ontologias de domínio de áreas de conhecimento e objetos de aprendizagem e seus metadados.

4 SISTEMAS BASEADOS EM AGENTES

Existem demandas específicas da educação em desenvolver sistemas capazes de tomar decisões de maneira autônoma, baseados em informações coletadas do ambiente educacional. Novas necessidades foram trazidas também pelo advento da Web Semântica (BERNERS-LEE et al. 2001) e de ambientes de *e-learning*. Há demanda pelo acesso a bases de dados heterogêneas e distribuídas, busca por conteúdos educacionais, e, ainda, a possibilidade de recomendação personalizada desses conteudos. Nesse contexto a tecnologia de agentes é promissora.

Agentes podem realizar diversas tarefas em ambientes de aprendizagem, tais como monitorar as atividades do estudante, capturar de forma automática as informações de contexto dinâmico do estudante, recomendar conteúdos de interesse deste, entre outras (RUSSEL; NORVING, 1995).

Diante do aumento no número de estudantes que interagem com os sistemas de suporte à aprendizagem, o uso de agentes para realizar estas tarefas torna-se extremamente importante, principalmente devido ao fato de serem tarefas complexas (RUSSEL; NORVING, 1995).

4.1 AGENTES

Existem na literatura muitas definições do que são agentes. No contexto da área de IA (Inteligência Artificial) o conceito de agentes inteligentes é muito explorado (RUSSEL; NORVING, 1995), (ANAND; GEORFEFF, 1995). No contexto de Web-Semântica são sistemas multi-agentes (ESCOBAR; LEMKE; BLOIS, 2006) com federações e/ou sociedades de agentes que interagem entre si para realizar determinada tarefa, ou mais simplesmente, agentes de software. A diferença básica entre um agente inteligente e um agente de software simples é que os agentes inteligentes são capazes de raciocinar sobre os elementos percebidos e escolher a melhor forma de agir de acordo com essa percepção (TAVARES, 2004).

Em Bresciani et al. (2004) é dito que cada vez mais o software precisa operar em diferentes plataformas, heterogêneas, sem recompilações, com o mínimo de informações sobre o ambiente operacional e os usuários. São aplicações robustas, com autonomia e que exigem pró-atividade por parte do software desenvolvido. Para que estas aplicações possam ser classificadas como agentes, devem possuir ainda outras características como: habilidade social, reatividade e pró-atividade além da já citada autonomia (RUSSEL; NORVIG, 1995),

(ANAND; GEORGEFF, 1995).

Um agente deve ser capaz de colaborar com o seu usuário para auxiliá-lo na realização de suas tarefas. Deste modo, o agente deve ser autônomo, lembrar as preferências dos seus usuários, possuírem um modelo de discurso, conseguir atingir seu objetivo, mesmo que tarefas menores não tenham sido atingidas, e colaborar com o usuário, conhecendo as experiências do mesmo (FONER, 1993).

Ferber e Gasser (2003) apud Deus (2012) define um agente inteligente como sendo uma entidade (real ou abstrata) que possui a capacidade de agir sobre ela mesma e sobre seu ambiente, podendo comunicar-se com outros agentes, e cujo comportamento é conseqüência de suas observações, de seu conhecimento e das interações com outros agentes. Ferber e Gasser (2003) apud Deus (2012) também destaca que um agente possui um comportamento autônomo (conseqüência de suas observações, de seu conhecimento e da suas intenções com os outros agentes).

Outro conceito relacionado a agentes é que são entidades computacionais criados com o propósito de atuarem de acordo com um conjunto de normas pré-definidas e de forma o mais autônoma possível. Entende-se então por agentes inteligentes como sendo um tipo de software com capacidade de interagir com seu ambiente e em tomar o seu conhecimento inicial, que pode estar presente em uma base de dados ou base de conhecimentos, como ponto de partida para a busca da realização dos seus objetivos no ambiente. Além disso, um agente inteligente comumente possui a capacidade de acumulação de experiências ou conhecimentos, que podem ser aplicados em situações similares no futuro.

Outros aspectos apresentados por Wooldrigde (1995) auxiliam na construção do perfil de um agente inteligente, entre elas se destacam, a autonomia como a capacidade do agente em trabalhar sem a necessidade da intervenção de usuários. A pró-atividade como a busca pelos seus objetivos no ambiente por iniciativa própria. A reatividade como instrumento de percepção e reação as mudanças do seu ambiente. A habilidade social como a capacidade de interagir com outros agentes no seu ambiente, através de algum tipo de comunicação. E por fim, a inteligência como sendo a presença de certo nível de conhecimento relacionado aos seus objetivos e a capacidade de assimilação de novos ao longo do seu ciclo de vida.

Para Tavares (2004), um agente pode funcionar de maneira autônoma e isolado na máquina do usuário ou em uma federação de agentes, construindo sistemas multi-agentes. A idéia fundamental, independente do tipo de agente, é a delegação. Ao software é delegada alguma autonomia de decisão. O agente ou um grupo de agentes recebe uma ou mais tarefas a serem executadas e através do reconhecimento do ambiente e da percepção do problema vai

em busca desta solução, realizando ações compatíveis de maneira autônoma deixando ao usuário somente o trabalho de requisitar a solução e aguardar por ela.

Em Santos (2006) algumas características importantes dos agentes são citadas como: adaptação (capacidade de aprendizado e através dele melhorar suas habilidades na resolução dos problemas que ele se propõe a resolver), aprendizado (se o agente é capaz de aprender sobre o ambiente que ele se encontra, ou sobre o(s) usuário(s) e outros agentes que interagem com ele), composição (composição de agentes no ambiente) e mobilidade (habilidade de migrar de um host para outro, perceber novos ambientes).

Já em Russel e Norving (1995), o agente é um objeto que pode ser visto como um perceptor dentro do ambiente que está inserido através de sensores e ainda um ator que age através de seus atuadores. Os agentes possuem seus próprios "*efectors*" para agir no seu ambiente. Além disto, há o conceito de racionalidade para agentes, onde um agente racional é classificado como tal, quando realiza uma ação que seja considerada correta dentro dos seus objetivos e metas. A Figura 15 ilustra a atuação do agente no ambiente.

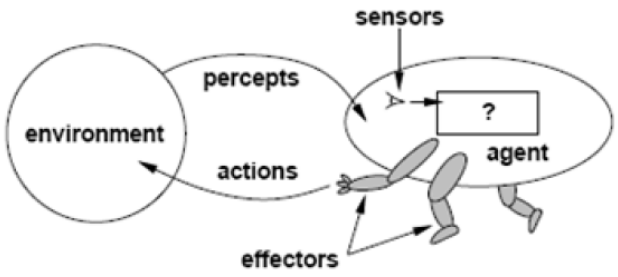


Figura 15: Interação do agente e o ambiente (RUSSEL; NORVIG, 1995)

O conceito de "fazer a coisa certa" é a ação realizada pelo agente que teve mais sucesso ao ser executada. Porém, é necessário decidir "como" e "quando" o sucesso do agente é avaliado. O "como" basicamente é a partir da ação executada pelo agente o quão próximo ele ficou do que era esperado. Já o "quando", é o sucesso avaliado dentro de um período de tempo, por exemplo, a medida do que foi alcançado dentro deste período.

Para Russel e Norving (1995), um agente racional ideal se traduz em: "Para cada seqüência possível de percepção, um agente racional ideal deve realizar a ação que maximizará a sua medida de performance, baseada em evidências providas pela sua percepção juntamente com o conhecimento nativo que o agente possui".

Outras características pertinentes aos agentes foram citadas em Tavares (2004) como:

- agentes podem operar sem controle direto de humanos ou de outros agentes, em outras palavras, são autônomos;

- agentes podem agir em sociedade com humanos e com outros agentes, eles se comunicam;
- agentes podem reagir a várias formas de estimulo dos domínios, eles são reativos;
- agentes podem por si mesmo tomar decisões para ajustar-se a metas definidas, eles são pró-ativos.

Agentes podem ser estruturados de várias formas. Podem ser agentes de software ou hardware, estáticos ou móveis, persistentes ou não-persistentes, reativos ou cognitivos (inteligentes). Uma das classificações mais importantes de agentes é em relação a eles serem reativos ou cognitivos.

Agentes reativos são agentes que selecionam ações a serem executadas com base exclusivamente na percepção atual, não levando em consideração o histórico de percepções. Uma vez que não possuem memória, são incapazes de planejar ações futuras (RUSSEL; NORVING, 1995). Por outro lado, os agentes congitivos são mais complexos, pois possuem uma representação explícita tanto do ambiente quanto de outros agentes. Este tipo de agente possui uma memória interna, o que possibilita planejar ações futuras baseando-se em situações que ocorrem anteriormente (RUSSEL; NORVING, 1995).

Há na literatura muitas classificações dos agentes baseados nos seus propósitos, na sua arquitetura, tecnologia, entre outras. Em Russel e Norving (1995) são apresentados diferentes papéis e aliado a cada papel a complexidade que um agente pode assumir.

Os agentes são separados em quatro tipos:

- agentes de reflexo simples (é o mais simples dos agentes, utilizando um conjunto simples de regras do tipo condição como base para tomada de decisão),
- agentes que rastreiam o mundo (requerem uma representação do mundo em estados possíveis para que os agentes possam tomar suas decisões),
- agentes baseados em metas (agentes são baseados em metas que devem ser alcançadas baseadas em uma seqüência de ações que devem ser tomadas) e
- agentes baseados na utilidade (onde as ações também são baseadas em metas, porém, contém heurísticas que possibilitam o agente "raciocinar" sobre qual caminho seguir caso haja mais de um possível para atingir determinada meta).

Já Nwana (1996), classifica os agentes em sete classes distintas:

- agentes colaborativos (ênfase na cooperação e autonomia),

- agentes de interface (ênfase em autonomia e aprendizado),
- agentes móveis (componentes de software capazes de navegar em uma rede como naWeb interagindo com servidores externos, buscando informações para seus usuários, entre outros),
- agentes de informação/Internet (utilizados em sites de busca),
- agentes reativos (agem sobre o princípio de estímulo/resposta respondendo as mudanças do ambiente o qual está inserido),
- agentes híbridos (possuem mais de uma característica em um único agente) e
- agentes inteligentes.

Um SMA (Sistema Multiagente), por sua vez, consiste de um conjunto de agentes autônomos que colaboram para resolver um problema o qual seria impossível solucionar com apenas um agente. Agentes de software podem realizar diversas tarefas em ambientes de aprendizagem, tais como monitorar as atividades do estudante, capturar dinamicamente informações do estudante, recomendar conteúdos de interesse deste, dentre outras atividades. Diante do aumento no número de estudantes que interagem com os sistemas de suporte à aprendizagem, o uso de agentes para realizar estas tarefas torna-se extremamente importante, principalmente devido ao fato de serem tarefas complexas para os facilitadores gerenciarem a distância.

4.2 MODELOS PARA A CONTRUÇÃO DE AGENTES E SEUS COMPORTAMENTOS

Na literatura podem ser encontrados diversos modelos para o desenvolvimento e documentação de agentes, entretanto destacamos a seguir dois modelos: BDI (ANAND; GEORGEFF, 1995) e 5C (VAN AART, 2005). O modelo BDI é o modelo clássico de desenvolvimento de agentes inteligentes e norteia o desenvolvimento de aplicações desde quando foi idealizado e deu origem, inclusive, a diversas metodologias exclusivas para o desenvolvimento de software baseados em agentes. O modelo 5C já possui aplicações comerciais desenvolvidas baseadas em seus conceitos e é, em parte, semelhante ao BDI.

4.2.1 MODELO BDI (*BELIEFS, DESIRES AND INTENTIONS*)

A suposição básica do modelo BDI é que ações são derivadas a partir de um processo chamado raciocínio prático, o qual é constituído de dois passos. No primeiro passo, deliberação (de objetivos), faz-se a seleção de um conjunto de desejos que devem ser

alcançados, de acordo com a situação atual das crenças do agente. O segundo passo é responsável pela determinação de como esses desejos concretos produzidos como resultado do passo anterior podem ser atingidos através do uso dos meios disponíveis ao agente (WOOLDRIDGE, 1999). As três atitudes mentais que compõem o modelo BDI são melhor detalhadas a seguir (Figura 20):

- **crenças** (*Beliefs*) – Informações sobre o ambiente, informativo. Representam as características do ambiente, as quais são atualizadas apropriadamente após a percepção de cada ação. Podem ser vistas como o componente informativo do sistema.
- **desejos** (*Desires*) – Metas para serem alcançadas, possivelmente com prioridades associadas a cada objetivo, motivacional. Contêm informação sobre os objetivos a serem atingidos, bem como as prioridades e os custos associados com os vários objetivos. Podem ser pensados como a representação do estado motivacional do sistema.
- **intenções** (*Intentions*) – A estratégia corrente escolhida para alcançar os objetivos, deliberativo. Representam o atual plano de ação escolhido. Capturam o componente deliberativo do sistema.
- **planos** *(Plans)* – Meios de atingir estados futuros do ambiente. Intuitivamente, os planos de ações são uma especificação abstrata tanto dos meios para atingir certos desejos (desires) quanto as opções disponíveis para os agentes. Cada plano tem um corpo descrevendo as ações primitivas ou sub-objetivos que precisam ser atingidos para o plano obter sucesso e o plano ainda possui uma condição de invocação que especifica os eventos que disparam a execução do plano e por último, existe uma condição de contexto que especifica em qual situação o plano é aplicável.

O processo de raciocínio prático em um agente BDI é apresentado na Figura 16.

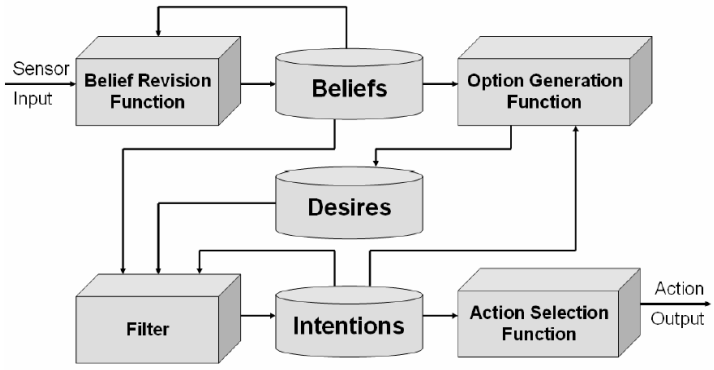


Figura 16: Diagrama de uma arquitetura *belief-desire-intention* genérica (WOOLDRIDGE,

1999)

Como a figura mostra, existem sete componentes principais em um agente BDI:

- conjunto de crenças (*beliefs*) atual, representando a informação que o agente tem sobre seu ambiente;
- função de revisão de crenças (*belief revision function*), a qual determina um novo conjunto de crenças a partir da percepção da entrada e das crenças do agente;
- função de geração de opções (*option generation function*), a qual determina as opções disponíveis aos agentes (seus desejos), com base nas suas crenças sobre seu ambiente e nas suas intenções;
- conjunto de opções (*desires*) corrente que representa os possíveis planos de ações disponíveis ao agente;
- função de filtro (*filter*), a qual representa o processo de deliberação do agente, que determina as intenções do agente com base nas suas crenças, desejos e intenções atuais;
- conjunto de intenções (*intentions*) atual, que representa o foco atual do agente, isto é, aqueles estados que o agente está determinado a alcançar;
- função de seleção de ação (*action selection function*), a qual determina uma ação a ser executada com base nas suas intenções atuais.

Existem várias metodologias de desenvolvimento de sistemas baseados em agentes que modelam os conceitos (ou pelo menos parte deles) do modelo BDI: TROPOS (BRESCIANI et al. 2004), AUML (*Agents Unified Modeling Language)* (ODELL et al., 2000), MESSAGE (EVANS et al. 2001), *Prometeus* (PADGHAM; WINIKOFF, 2003), entre outras.

4.2.2 MODELO 5C

O modelo 5C é um *framework* conceitual baseado na generalização dos diferentes tipos de capacidades dos agentes que podem ser encontrados na literatura. Este modelo foi criado com o intuito de servir como um guia para o projeto de desenvolvimento de aplicações baseadas em agentes inteligentes tanto protótipos quanto comerciais (VAN AART, 2005).

O desenvolvimento destas aplicações mostrou que as restrições funcionais e técnicas podem ser representadas ao longo de cinco dimensões, usando a noção de separação de conceitos. Este modelo 5C pode ser dividido em cinco dimensões: competência, comunicação, autonomia, planejamento e ambiente. Estas dimensões são enquadradas em

modelos onde cada modelo é responsável por um tipo particular de funcionalidade que o agente exige:

- competência: um agente é projetado para executar uma ou um pequeno número de tarefas. Podemos enxergar uma tarefa como a oferta de um serviço. Este agente pode executar suas tarefas que já fazem parte da sua própria natureza, ou, ainda, tem a possibilidade de consultar um sistema legado, por exemplo, para que possa executar seus serviços.
- autonomia: um agente tem uma identidade incluindo a descrição do seu nome, objetivo, estado atual, papel organizacional, capacidades que possui e localização computacional. Mesmo que um agente se mova de um servidor para outro é necessário assumir que ele ainda continua sendo o mesmo agente. Complementando seu comportamento e autonomia, o agente precisa ter as seguintes capacidades: introspecção, reflexão, gerenciamento do seu ciclo de vida e um plano de instruções.
- planejamento: o planejamento está ligado à autonomia dos agentes. A autonomia de um agente é o que torna um agente de software inteligente tão diferenciado (GASPARI; MOTTA, 1994), (WOOLDRIDGE, 2009). Enquanto um sistema especialista é projetado para realizar uma tarefa de maneira racional, ancorado em um ambiente fixo que determina o comportamento do processo, a racionalidade de um agente é estendida para um número de decisões a serem tomadas: como fazer, quando fazer, com qual freqüência e quanto tempo levará.
- ambiente de execução: os agentes de software estão localizados em um ou mais ambientes (NWANA, 1996) onde eles podem interagir com outros atores como: outros agentes, pessoas, sistemas legados e fontes de informação.

4.3 DESENVOLVIMENTO DE SISTEMAS MULTIAGENTES

Não há nenhum consenso sobre uma tecnologia específica para o desenvolvimento de agentes. Nas atividades conceituais que lidam com o ciclo de vida de desenvolvimento, por exemplo, muitas metodologias foram propostas como *AUML* (ODELL et al. 2000), *MaSE* (WOOD; DELOACH, 2000), TROPOS (BRESCIANI et al. 2004), *Prometeus* (PADGHAM; WINIKOFF, 2003), *MAS-CommonKADS* (IGLESIAS; GARIJO, 2005) apud (MENDES, 2009), Gaia (WOOLDRIDGE et al. 2000), entre outras.

No desenvolvimento das aplicações baseadas em agentes, algumas questões devem ser ressaltadas, pois, apesar de ter um ciclo de vida semelhante ao de um software convencional, pontos adicionais devem ser verificados durante todo o processo.

É necessário haver alguma metodologia que modele o papel dos agentes dentro do software, como os agentes se relacionam com as outras partes do sistema, para que tipos de estrutura de dados esses papéis serão mapeados, modelos comportamentais também são importantes e na implementação propriamente dita, qual paradigma será utilizado, qual *framework* de desenvolvimento, por exemplo, além de saber quais protocolos de comunicação e formato de mensagens serão usados para a interação entre agentes e entre os agentes e o software.

O NEnC (Núcleo de Engenharia do Conhecimento), cujas pesquisas incluem essa dissertação, adotou em trabalhos anteriores, como Mendes et al. (2011), a metodologia Ingenias (GÔMES-SANZ; FUENTES, 2009) para a modelagem e representação dos agentes, dessa forma esse trabalho segue a opção do grupo e a descreve a seguir.

4.3.1 METODOLOGIA INGENIAS

O grupo de pesquisa GRASIA (http://grasia.fdi.ucm.es) desenvolveu a metodologia INGENIAS (*Engineering for Software Agents*), (GÔMES-SANZ; FUENTES, 2009), a partir de uma análise das metodologias de desenvolvimento de sistemas multi-agentes já existentes, onde verificou que nem todos os aspectos eram cobertos no momento de desenvolver a aplicação. Entretanto, as fases de extração dos objetivos, definição e verificação dos protocolos de comunicação ou execução das tarefas, por exemplo, eram contemplados em parte pelas metodologias analisadas.

A INGENIAS foi baseada em conceitos da engenharia como a definição de *workflows*, encapsulamento das funcionalidades através de papéis, grupos e organizações, e ainda, a possibilidade de modelar diferentes visões do sistema tanto estruturais como funcionais. A metodologia leva em consideração todo o ciclo de vida do desenvolvimento de um software: análise, projeto e implementação.

A INGENIAS faz reuso de uma metodologia já existente MESSAGE/UML (EVANS et al. 2001), escolhida por apresentar uma visão bem próxima a da engenharia no que diz respeito ao projeto de um sistema multi-agente. A MESSAGE/UML define cinco tipos de visões do sistema baseadas em uma linguagem de meta-modelagem, visões estas, aplicadas ao longo de todo o ciclo de desenvolvimento para representar os aspectos concretos do sistema. A INGENIAS além de reusar as visões existentes na MESSAGE/UML, também a estende

com uma nova visão (visão do ambiente), reconstrói as visões para atender ao modelo BDI (*Belief Desire Intention*), provê ferramentas de modelagem, documentação do sistema e geração de código automática dos agentes para o projeto.

A abordagem geral da metodologia descrita para especificar um SMA (Sistema Multi-Agente) é dividir o problema em aspectos mais concretos, que juntos formam diferentes visões do sistema. A organização do SMA é do tipo comunidade de especialistas, pois cada um dos tipos de agentes criados encontra-se no mesmo nível, sendo cada um deles especialista em determinada tarefa (PAVÓN et al. 2005). Os agentes interagem entre si através de um protocolo de comunicação ACL (*Agent Communication Language*) previamente estabelecido. A diferença para outras metodologias é como as visões são construídas e como são integradas no desenvolvimento do SMA. Para que as diversas visões sejam construídas e os modelos que as representam sejam gerados, uma linguagem de meta-modelo é utilizada, GOPRR (*Graph, Object, Property, Relationship, and Role*) (LYYTINEN; ROSSI, 1996) apud (MENDES, 2009). Na metodologia INGENIAS existem cinco meta-modelos que originaram as cinco visões para o desenvolvimento de um SMA (Figura 17).

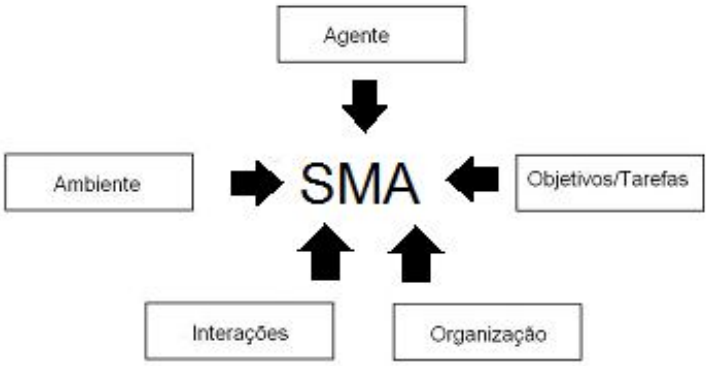


Figura 17: As cinco visões para SMA definidas com INGENIAS

MODELO DE ORGANIZAÇÃO

Segundo Filippetto (2007) o modelo de organização descreve como componentes do sistema (agentes, papéis, recursos e aplicações) são agrupado, quais tarefas são executadas em comum e quais objetivos são compartilhados. O modelo de organização se assemelha aos diagramas de arquitetura para sistemas orientados a objetos, visa descrever a organização e como ocorrem os fluxos de trabalhos definidos para a organização.

Segundo Mendes (2009) a partir da metodologia *MESSAGE/UML* (EVANS et al. 2001) três pilares foram construídos para dar suporte aos modelos de organização:

- distinção entre agentes e organizações: as organizações são compostas por

agentes. As mesmas só existem mediante a existência de agentes que as compõem. Porém, os agentes podem existir sem as organizações. Como nas organizações humanas, os agentes sempre são os responsáveis finais por executar tarefas dentro de uma organização;

- a organização delega sua estruturação em uma estrutura organizacional: não há relações de agregação entre as organizações. Existem diferenças entre a organização e a estrutura organizacional. A organização tem um conjunto de atributos que não existem na estrutura organizacional. O ciclo de vida também é diferente. As estruturas organizacionais se criam e se destroem dentro de uma organização sem que esta deixe de existir. Os agentes e os recursos sempre pertencem à organização apesar de serem disponibilizados dinamicamente a diferentes estruturas organizacionais. Por último, se uma organização é destruída, consequentemente todas as suas estruturas organizacionais também o são;
- fluxo de trabalho: permitem contextualizar a execução das tarefas e interrelacioná-las umas com as outras, independente de quem são os executores da mesma.

MODELO DE AGENTES

Este modelo descreve o agente, suas tarefas, objetivo, estado mental e papeis.

Segundo Filippetto (2007) o modelo de agentes visa descrever os agentes que serão utilizados no sistema, juntamente com seus objetivos, tarefas e seus estados mentais. Além do estado inicial é possível representar neste modelo os estados intermediários do agente a ser modelado. Um agente é considerado uma entidade autônoma que possui tarefas e busca os objetivos definidos para si. Para atingir o objetivo, os seguintes aspectos são modelados e analisados:

- responsabilidades: representadas pelas tarefas que o agente saberá executar e dos objetivos que se compromete a alcançar. Usa-se geralmente o termo role (papel) para agrupar as funcionalidades e as propriedades que aparecem com frequência no projeto do agente;
- comportamento: este termo denota várias interpretações nas diferentes metodologias existentes. A UML avalia comportamento como um conjunto de chamadas a procedimentos, *MaSE* e *MESSAGE* a troca de mensagens entre os agentes e até transições em máquinas de estado são avaliadas como

"comportamento" de um agente.

MODELO DE TAREFAS E OBJETIVOS

Este modelo descreve as relações entre objetivos e tarefas e suas estruturas e expressa as entradas e saídas das tarefas e seus efeitos sobre o ambiente e o estado mental do agente.

Segundo Mendes (2009) este modelo tem como objetivo principal representar as motivações do SMA, definir as ações identificadas nos modelos de organização, interações e agentes e como afetam estas ações e suas responsabilidades. Através deste modelo, têm-se parte da especificação em alto nível de como será o controle do agente. Trata-se de poder expressar quais são as conseqüências de executar as tarefas e o porquê de executá-las.

Nesta metodologia, a tarefa tem dois papéis principais: transformar o estado global e de processo. A primeira está relacionada à representação das pré-condições e pós-condições dos agentes que permite o planning e tarefas de raciocínio. Como processo, ela representa uma estrutura pragmática e direta: um conjunto de instruções que devem ser executadas.

Os objetivos na metodologia são vistos como alternativas que se apresentam aos agentes em um dado momento. Para representá-las existem duas tendências:

- objetivo como agregação: é o enfoque predominante. O objetivo é visto como uma descrição do estado do mundo que deve ser alcançado. Quando a descrição do estado do mundo se faz com um conjunto de predicados, o objetivo se converte em uma agregação destes predicados;
- objetivos como entidades: é o enfoque usado no clássico BDI. Não se trata de agregações de predicados e sim de entidades auto-representativas. Seguindo o modelo BDI, trata-se de um desejo que queira satisfazer.

MODELO DE INTERAÇÃO

Este modelo descreve cada interação, inclui os atores envolvidos, objetivos, e uma descrição do protocolo que se segue a interação e diagramas UML de interação e diagramas INGENIAS (GÔMES-SANZ; FUENTES, 2009).

Segundo Mendes (2009), o papel das interações dentro de um SMA é fundamental pois identifica dependências entre os componentes do sistema e contribuem para a especificação do comportamento dos mesmos assim como a funcionalidade associada. As interações determinam o comportamento dos agentes exibindo a reação durante suas atuações dentro do SMA. Como o comportamento é em função dos objetivos e tarefas a executar, fica explícito a ligação entre as interações e os objetivos e tarefas de cada agente. A especificação

completa de uma interação deve cobrir os seguintes itens:

- atores: um ator deve mostrar porque está participando da interação. É coerente com um modelo de agente que se baseia no princípio da racionalidade;
- unidade de interação: a natureza da unidade de interação determina como o receptor deve ser processado. Pode ser simples como um passo da mensagem como pode ser complexa por adequar as interações de acordo com as capacidades do receptor;
- ordem sobre as unidades de interação: as unidades de interação se organizam seguindo um protocolo padrão adequado para uma situação concreta. Este protocolo é responsável em organizar a interação e sua execução.

A especificação completa de uma interação deve também apresentar as ações executadas na interação, são elas:

- critérios para decidir quando executar uma tarefa: para executar uma tarefa não é simplesmente o agente aceitá-la. Ele precisa ter autonomia de negar a solicitação, avaliar se pode abandonar uma tarefa já em execução e assumir estas novas ou avaliadas estas situações, assumir efetivamente a tarefa;
- consequências da execução de uma tarefa: ao final da execução de uma tarefa esperam-se mudanças no estado do agente e do ambiente no qual ele está inserido;
- definição do contexto: o contexto consiste em mostrar o que ocorre com o sistema quando se inicia uma interação. Esta informação demonstra qual o estado necessário dos agentes para que sejam acionados durante a interação, indicando os atores participantes, os motivos que levaram estes agentes a participarem, e que objetivos são perseguidos pela interação;
- modelo de controle: O controle assegura que a interação vai ser realizada da maneira que foi definida. Este controle deverá levar em consideração que várias interações podem ser realizadas em paralelo, assumindo assim um papel central na coordenação das interações.

MODELO DE AMBIENTE

O Modelo de Ambiente define a percepção do agente em termos dos elementos do sistema e identifica também os recursos do sistema responsáveis pela gestão do agente.

Os recursos podem ser qualquer objeto do ambiente que possua uma funcionalidade concreta, que são indispensáveis para a execução de uma tarefa, o recurso é apenas utilizado

para se alcançar um dos objetivos. Já os agentes são representados neste diagrama para demonstrar seus relacionamentos em relação aos elementos possíveis do ambiente (GRUPO GRASIA, 2013). Segundo Mendes (2009), os mecanismos de percepção dos agentes são funções diretas de como se descrevem as entidades do sistema.

A evolução do ambiente é a soma da evolução de seus componentes (recursos, entidades e agentes) influenciada pelas dependências entre eles (uma ação sobre um determinado recurso pode afetar outras entidades). Em suma, o modelo define a percepção dos agentes em termos dos elementos existentes no SMA. Também identifica os recursos do sistema e os responsáveis pelo seu gerenciamento.

IDK (INGENIAS DEVELOPMENT KIT)

O IDK (*INGENIAS Development Kit*) (Figura 18) é uma ferramenta que pretende facilitar o desenvolvimento de MAS (Sistemas Multi-Agentes), apoiando o processo de desenvolvimento INGENIAS.

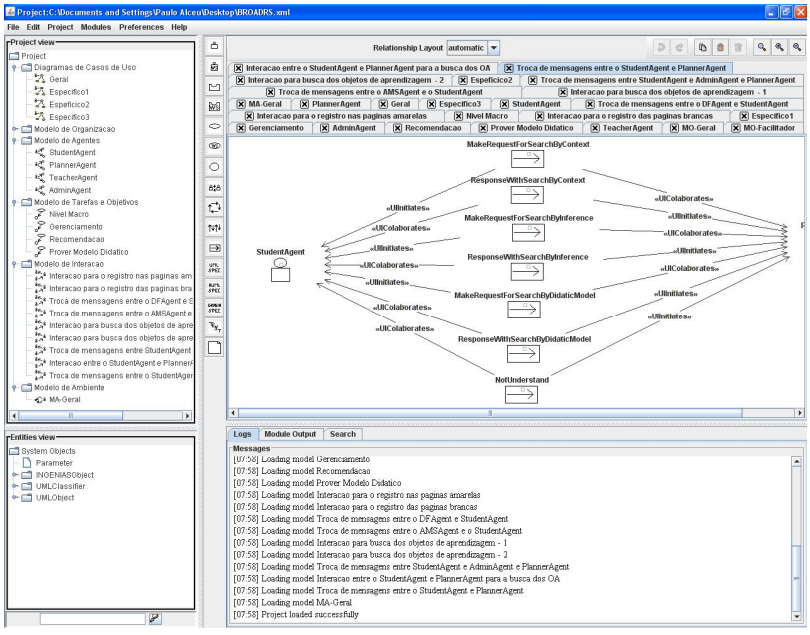


Figura 18: Ambiente IDK - INGENIAS Development Kit

O IDK apresenta notação visual para criação de seus modelos com símbolos (Figura 19) e define símbolos para modelagem como, por exemplo, dois símbolos de agente interagindo com duas setas para representar a interação entre eles.

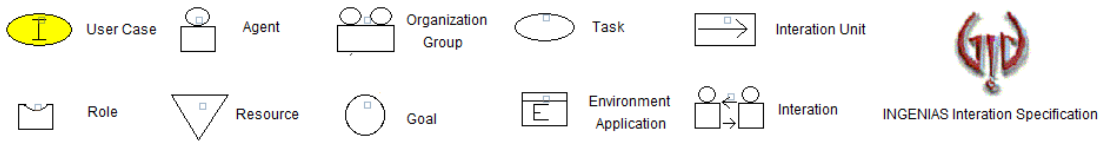


Figura 19: Elementos relevantes da notação INGENIAS

O IDK também implementa técnicas de notação e documentação das fases de desenvolvimento e gera codificação automatizada a partir de uma linguagem de modelagem específica para JADE, que é uma linguagem de programação orientada a agente (Figura 20).

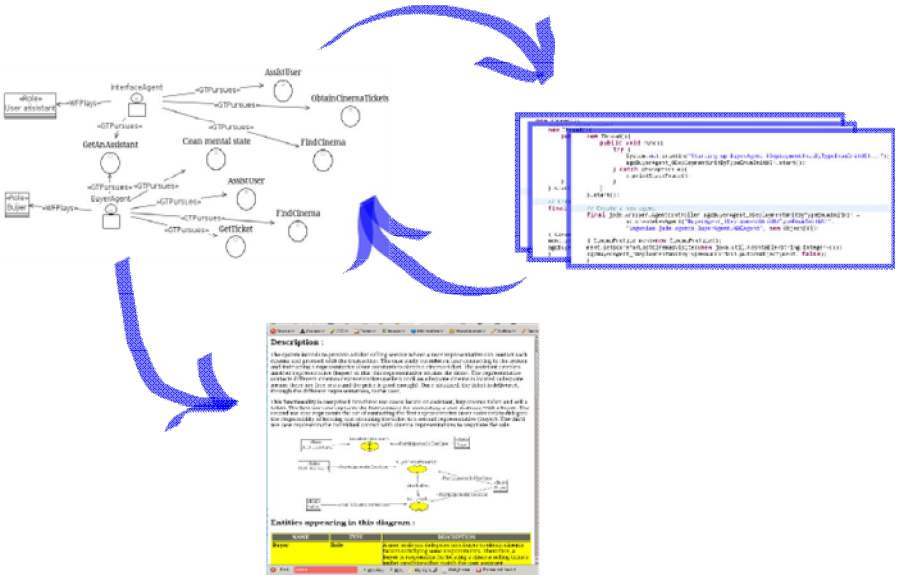


Figura 20: *INGENIAS Development Kit (IDK) (http://grasia.fdi.ucm.es)*

4.4 *FRAMEWORK* PARA AGENTES

4.4.1 JADEX

Jadex (POKAHR; BRAUBACH, 2007) é um mecanismo de raciocínio orientado a agentes, no qual agentes racionais são escritos em XML e na linguagem de programação JAVA (GOSLING, 2011). Um dos aspectos principais do Jadex é que ele não apresenta uma nova linguagem de programação. Ao invés disso, os agentes Jadex podem ser programados nos ambientes de desenvolvimento integrados orientados a objetos.

Outro conceito importante é a independência do Jadex. Como ele é fracamente acoplado com o *middleware* sobre o qual ele é executado, Jadex pode ser usado com diferentes plataformas de agentes. Atualmente, dois adaptadores estão disponíveis. O primeiro é a plataforma JADE (BELLIFEMINE, 2003), a qual é bem conhecida e de código livre. O segundo é o adaptador *Jadex Standalone*, o qual é um ambiente pequeno mas rápido.

Jadex segue o modelo BDI, utilizando crenças, objetivos e planos como objetos de primeira classe, que podem ser criados e manipulados dentro do agente. No Jadex, agentes têm crenças, que são armazenadas em uma base de crenças. Objetivos representam motivações concretas, como por exemplo, estados a serem atingidos, e influenciam no comportamento do agente. Para atingir seus objetivos, o agente executa planos, os quais são roteiros procedurais codificados em JAVA (GOSLING, 2011). A arquitetura abstrata de um agente Jadex é apresentada na Figura 21.

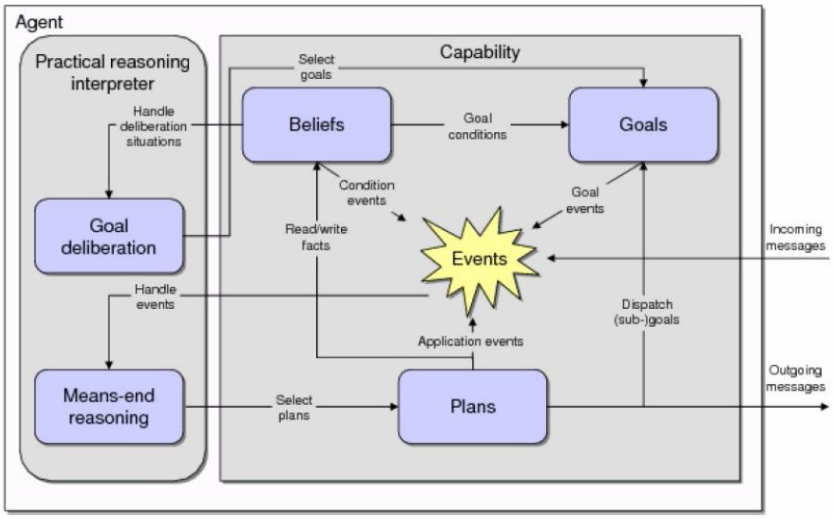


Figura 21: Arquitetura abstrata Jadex (POKAHR; BRAUBACH, 2007)

Os principais componentes do Jadex são apresentados com mais detalhes a seguir:

- *capability:* capacidade permite que crenças, planos e objetivos sejam colocados em um módulo de agente. Este módulo pode ser reusado sempre que necessário. Capacidades podem conter subcapacidades formando um hierarquia arbitrária de módulos.
- *beliefs:* crenças representam o conhecimento do agente sobre o seu ambiente e sobre si mesmo. Em Jadex, as crenças podem ser qualquer objeto JAVA. Elas são armazenadas em uma base de crenças e podem ser referenciadas em expressões, bem como ser acessadas e modificadas por planos usando a interface da base de crenças.
- *goals*: objetivos compõem a postura motivacional do agente, é o que orienta suas ações. Diferentemente dos sistemas BDI tradicionais, os quais tratam objetivos como um tipo especial de evento, objetivos são um conceito central no Jadex, o qual segue a idéia geral que objetivos são desejos concretos e momentâneos de um agente. Para qualquer objetivo que ele tenha, um agente

irá engajar-se em ações apropriadas, até que ele considere o objetivo como atingido, inatingível ou não mais desejado.

- *plans*: planos representam a forma como o agente atuará em seu ambiente. Entretanto, os planos pré-definidos pelo desenvolvedor compõem uma biblioteca de ações que o agente pode executar. Dependendo da situação corrente, planos são selecionados em resposta à ocorrência de eventos ou de objetivos. A seleção de planos é feita automaticamente pelo sistema e representa um aspecto principal da infraestrutura BDI.
- *events:* uma importante propriedade dos agentes é a capacidade de reagir a diferentes tipos de eventos. Jadex suporta dois tipos de eventos a nível de aplicação, os quais podem ser definidos pelo desenvolvedor. Eventos internos podem ser usados para denotar uma ocorrência dentro de um agente, enquanto eventos mensagens representam uma comunicação entre dois agente ou mais. Eventos normalmente são tratados por planos.

O raciocínio no Jadex é um processo que consiste de dois componentes entrelaçados. Por um lado, o agente reage a mensagens que chegam, a eventos internos e a objetivos através da seleção e execução de planos. Por outro lado, o agente continuamente delibera seus objetivos correntes, para decidir a respeito de um subconjunto consistente, o qual deve ser perseguido.

O interpretador consiste de uma agenda de componentes armazenando meta-ações a serem executadas. O modo básico de operação é simples: o agente seleciona uma meta-ação de sua agenda e a executa quando as pré-condições da ação são satisfeitas. Caso contrário, a ação é simplesmente descartada. A execução da ação pode produzir ações posteriores, que são adicionadas à agenda seguindo uma estratégia de inserção customizável. Atualmente, a estratégia de inserção distingue ações relacionadas e não relacionadas, onde ações relacionadas são adicionadas como nodos filho do nodo atual.

Além da criação de novas entradas na agenda, a execução de ações pode ter efeitos colaterais que são de importância para o agente, tal como quando uma crença é alterada ou um objetivo é descartado. Essas ocorrências são capturadas como eventos de sistema e podem causar alterações nele. Elas são computadas por um componente de determinação de alteração correspondente. Para determinar quais efeitos que um evento de sistema tem, o componente avalia as condições impactadas. Se uma nova condição é disparada, novas ações são produzidas e adicionadas à agenda.

O mecanismo é especificamente projetado para a construção de sistemas que utilizam

princípios e práticas de engenharia de software existentes e ficam como uma camada independente que pode ser flexivelmente implantada em plataformas middleware, como por exemplo JADE.

Jadex possui uma série de ferramentas disponíveis. Elas permitem o controle da execução dos agentes, verificando uma série de dados, tais como suas crenças, objetivos e planos, depuração e documentação. Além disso, é fornecido um plugin para a ferramenta Protégé, que serve para o projeto de ontologias.

4.4.2 JADE

No que diz respeito às plataformas de desenvolvimento, protocolos de interação e outros assuntos relacionados à implementação dos agentes, uma iniciativa tem tido maior destaque na literatura: a plataforma JADE (*Java Agent Development Framework*) (BELLIFEMINE, 2003). O *framework* JADE é totalmente implementado em JAVA (GOSLING, 2011). Ele simplifica o processo de implementação de sistemas multiagentes através de um *middleware* que está de acordo com as especificações FIPA e através de um conjunto de ferramentas gráficas que suportam o *debugging* e a implementação. A plataforma de agentes pode ainda ser distribuída entre várias máquinas (que não necessariamente sejam do mesmo sistema operacional) e o controle da configuração de cada máquina pode ser feita remotamente via interface gráfica. Os agentes podem ainda mover-se entre as diversas máquinas quando e como for requerido. Assim os agentes são executados diretamente na plataforma JADE, e esta por sua vez é executada na plataforma JAVA, que deve estar devidamente instalada e funcionando perfeitamente.

Os princípios básicos sobre os quais a plataforma foi desenvolvida são:

- interoperabilidade: JADE atende as especificações FIPA, portanto, os agentes JADE podem se relacionar com agentes de outras plataformas desde que estejam definidos ou implementados no mesmo padrão;
- uniformidade e portabilidade: JADE provê um conjunto de APIs homogêneas que são independentes da rede e da versão da linguagem JAVA;
- facilidade de uso: a complexidade da plataforma está oculta através de APIs simples e intuitivas de se manipular.

Além dos princípios básicos, a plataforma JADE inclui três elementos principais (BELLIFEMINE, 2003): (1) conjunto de pacotes que proporcionam uma série de componentes para o desenvolvimento dos agentes; (2) um ambiente de execução que fornece

os serviços básicos necessários para a execução dos agentes; e (3) um conjunto de ferramentas gráficas que permitem administrar e monitorar a atividade dos agentes ativos do sistema. Cada instância do ambiente de execução chama-se container e oferece uma serie de serviços de acordo com a especificação FIPA, como o registro, o sistema de troca de mensagens e o serviço de autenticação. Em um mesmo ambiente ou plataforma, podem existir diversos containeres, porém, todo ambiente terá sempre um container principal onde os outros containeres deverão se registrar. A Figura 22 exibe esta arquitetura.

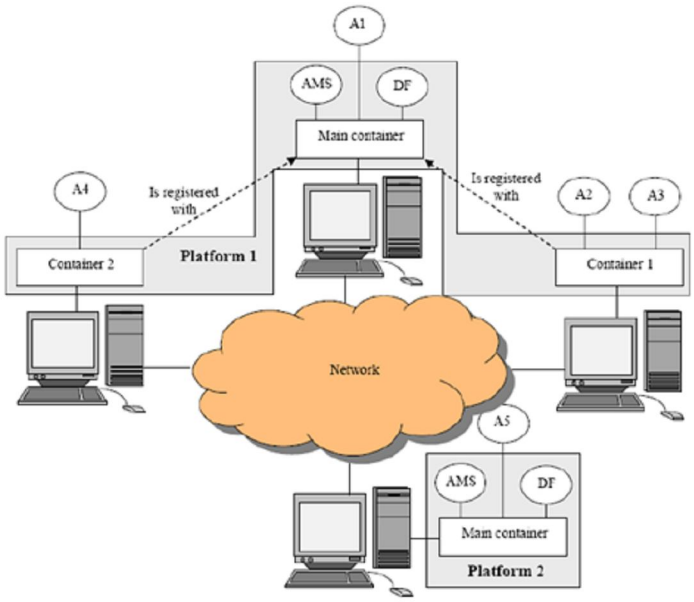


Figura 22: Plataformas e containeres no JADE 4.1

Ainda pertencentes à plataforma JADE e contidos sempre nos containeres principais de cada instância em execução, existem dois agentes específicos, sempre inicializados de forma automática:

- AMS (*Agent Management System*): proporciona um serviço de páginas brancas, assegurando-se que todo agente dentro da plataforma tenha um nome único. Ele representa a autoridade na plataforma supervisionando o acesso e o uso da mesma;
- DF (*Directory Facilitator*): proporciona um serviço de páginas amarelas, onde este agente pode encontrar outros agentes através dos serviços que realizam.

Podemos dividir em dois os tipos de modelos de comunicação entre os agentes (LUCENA, 2003). A primeira delas é a arquitetura do quadro negro onde os agentes se comunicam entre si de maneira indireta através de um quadro negro (BUSHMANN et al. 1996). O quadro negro é uma estrutura de dados persistente, onde existe uma divisão em

regiões ou níveis para facilitar a busca pela informação desejada. Ele funciona exatamente como o meio de interação entre os agentes, funcionando como uma espécie de repositório de mensagens. Os agentes escrevem as mensagens no quadro e ao mesmo tempo lêem mensagens escritas por outros agentes.

O segundo tipo de comunicação entre agentes é utilizando troca de mensagens diretas entre eles. Para que este paradigma seja colocado em prática baseada no padrão FIPA, (FIPA, 2004), utilizado para desenvolvimento da comunicação entre agentes, alguns detalhes precisam existir na plataforma. Primeiro a funcionalidade de *whitepages*, que basicamente funciona como um serviço de encontrar os agentes pela sua identificação, como uma lista telefônica normal onde buscamos o telefone de alguém através do nome da pessoa.

E outra funcionalidade importante é o serviço de *yellowpages*, onde os agentes localizam outros agentes não pela identificação, mas pelos serviços que eles podem prover dentro da plataforma.

Todo este paradigma é sustentado pelos chamados agentes facilitadores, que são os responsáveis pelos endereços (identificação) dos agentes e seus respectivos serviços ou em nomes mais específicos no padrão FIPA temos um DF (*Directory Facilitator*), responsável pelo serviço de *yellowpages*, e um MAS (*Agent Management System*), responsável pelas whitepages.

O modelo arquitetural da plataforma JADE pode ser visto na Figura 23.

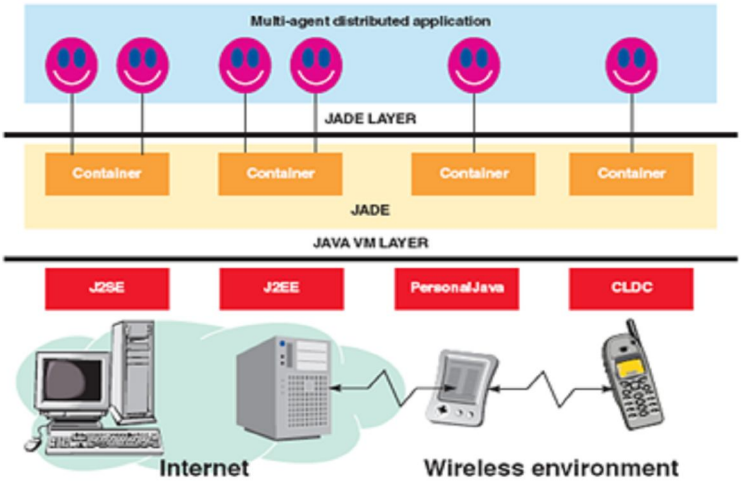


Figura 23: Arquitetura da plataforma JADE (BELLIFEMINE, 2003)

São oferecidas tanto as bibliotecas necessárias para desenvolver os agentes da aplicação quanto o ambiente *run-time* que provê os serviços como *yellowpages* e *whitepages*, por exemplo, e que precisam estar ativos no dispositivo onde os agentes irão ser executados. A plataforma deve prover estes serviços e trabalhar com uma linguagem específica de

comunicação entre os agentes ACL *(Agent Communication Language*). Cada instância da plataforma JADE de *run-time* é chamada container onde os agentes são criados e hospedados.

Outra característica interessante do ambiente JADE é atuar como um middleware, ou seja, proporcionar aos agentes que estão ali executando uma camada genérica que lhes abstrai quaisquer questões relativas a hardware e software, tendo apenas a necessidade da plataforma JAVA estar executando naquela máquina. Essa característica permite aos programadores apenas ter o conhecimento das estruturas necessárias ao desenvolvimento dos agentes e utilização dos mecanismos oferecidos pela plataforma.

Esse recurso do JADE converte as informações contidas no objeto JAVA na forma de uma expressão, e disponibiliza funções para a extração destas informações e inserção em um novo objeto JAVA. Essa expressão é no formato de uma *string*, ou seja, uma sequência de caracteres que conterá todas as informações do perfil do usuário que o agente representa.

Cada agente dentro da plataforma é identificado por um identificador único e pode prover um conjunto de serviços. Ele pode modificar ou registrar seus serviços e ainda procurar por outros agentes que provêem os mesmos serviços. Ele mesmo controla seu ciclo de vida, e em particular, se comunica com outros agentes numa arquitetura ponto a ponto (BELLIFEMINE, 2003).

4.5 CONSIDERAÇÕES FINAIS DO CAPÍTULO

O uso de agentes no desenvolvimento de aplicações faz parte dos mais diversos campos de pesquisa no contexto da Ciência da Computação, oferecendo recursos interessantes principalmente no desenvolvimento de aplicações distribuídas e complexas e ainda em domínios que possuem uma composição bastante heterogênea dos seus componentes e onde existam interesses que estejam em conflito (ELAMY, 2005).

Os tipos de agentes implementados na arquitetura BROAD-RS são agentes baseados na reatividade (por reagir a estímulos/mensagens recebidas de outros agentes) com habilidade de socializar pois os agentes verificam as interações dos professores e alunos no sistema, como atualização de dados de contexto e plano didático. Os agentes também verificam a interação dos dados com a rede de ontologias e serviços da arquitetura para realizar recomendações de OA de maneira mais dinâmica e ágil. Um agente pró-ativo chamado de Admin também é implementado para monitorar a disponibilidade dos agentes e seus serviços.

Com a utilização da metodologia INGENIAS é possível diminuir a dificuldade encontrada em se aplicar metodologias de SMA (Sistemas Multi-Agente) em problemas reais, reduzir a dimensão da aplicação, cada vez maior em sistemas atuais, aumentar a capacidade

de incorporar as práticas da engenharia de software no desenvolvimento de SMA e.permitir uma construção incremental das especificações de SMA, assegurando correto desenvolvimento nas fases de análise, projeto e implementação em um ciclo de vida.

A metodologia INGENIAS também tem como vantagem o ferramental disponibilizado e sua integração com a plataforma de desenvolvimento de agentes JADE (*Java Agent Development Framework*) (BELLIFEMINE, 2003), gerando código a partir dos modelos criados na utilização da metodologia. Outro ponto importante a ser considerado é a modelagem do ambiente, onde são modeladas em detalhes as interfaces dos agentes com o ambiente externo. Este aspecto é particularmente interessante para o BROAD-RS, uma vez que utiliza repositórios semânticos interagindo diretamente com agentes e fazendo o papel destes em alguns casos.

Também escolhemos o *framework* JADE pois é de código aberto e, segundo García (2007), conta com um grande número de colaboradores tanto da indústria quanto da academia, levando a numerosos projetos baseados nesta plataforma, tornando-o um projeto mais maduro e, consequentemente, levando à sua escolha para este trabalho.

O padrão de estruturação específico ao ambiente do servidor de aplicações, foi implementado utilizando um suporte a ontologias e conteúdos de linguagem que a própria plataforma JADE disponibiliza. Os mecanismos oferecidos pela plataforma possibilitam a implementação conforme a necessidade do sistema. Ela fornece interoperabilidade, já que é implementada de acordo com o padrão FIPA, sendo assim, é possível até mesmo se comunicar com outros agentes de outras plataformas se necessário. Oferece portabilidade, pois possui um conjunto de APIs para vários tipos de rede, dispositivos móveis como celulares, PDAs, entre outros, aumentando assim a gama de aplicações que podem ser desenvolvidas. A versão utilizada no desenvolvimento do BROAD-RS foi a JADE 3.5.

5 TRABALHOS RELACIONADOS

Neste capítulo são apresentadas duas visões de trabalhos relacionados. A primeira parte apresenta os modelos e propostas de recomendação de conteúdos educacionais com análises dos tipos de sistemas, tecnologias, recursos, metodos e características, considerando que a utilização de sistemas sensíveis ao contexto tem sido proposta na literatura com o objetivo de recomendar conteúdos de objetos de aprendizagem. A segunda parte apresenta uma visão dos trabalhos relacionados dando ênfase nas catacterísticas em comum dos modelos, em função das diferentes perspectivas dos autores.

5.1 MODELOS E PROPOSTAS DE RECOMENDAÇÃO

Jácome Júnior, Mendes Neto e Silva, (2012) apresentam um ambiente chamado *MobiLE* capaz de realizar a recomendação sensível ao contexto de objetos de aprendizagem, definidos em um formato padrão, através do uso de ontologias de um sistema multi-agente e da aplicação de um algoritmo genético.

Zaíne (2002) propõe um agente de recomendação que modela representações do perfil de seus usuários através do uso de técnicas de mineração sobre o histórico de acesso dos mesmos. A partir desta representação o agente realiza recomendações baseadas no que usuários com históricos semelhantes fizeram em determinada atividade didática.

Zhuhadar et al. (2009) realiza o processo de recomendação baseado em buscas de conteúdos educacionais realizadas por usuários. O perfil de seus usuários considera áreas de conhecimento associadas a estes de maneira a comparar com as áreas dos conteúdos educacionais possibilitando assim o processo de recomendação. Para a descrição destas áreas é proposto o uso de ontologias.

O trabalho de García et al. (2009) apresenta um *framework* para recomendar alternativas que auxiliem professores a prover formas didáticas que auxiliem seus alunos na resolução de tarefas propostas em sala de aula. Para tanto, baseia-se em métodos de Filtragem Colaborativa apoiados em regras de associação extraídas automaticamente através de técnicas de mineração de dados dos históricos de ações no sistema.

Em Primo, Vicari e Silva (2010) é apresentado um modelo para a recomendação de conteúdos educacionais descritos através de metadados. Esse modelo considera perfis de usuários e o suporte a interoperabilidade entre aplicações educacionais, além de aspectos cognitivos de aprendizado. Este trabalho também apresenta como conteúdos educacionais

podem ser descritos através de ontologias, o que facilita também a inferência dos conteúdos apropriados aos perfis dos usuários.

Em Gluz e Vicari (2010) é apresentada a MILOS (*Multiagent Infrastructure for Learning Object Support*), uma infraestrutura combinando ontologias e agentes, que implementa as funcionalidades necessárias aos processos de autoria, gerência, busca e disponibilização de OAs compatíveis com a proposta de padrão de metadados brasileiro de OAs OBAA (Objetos de Aprendizagem Baseados em Agentes) (VICARI et al. 2009).

Ferro et al. (2011) aborda a criação de um modelo de sistema de recomendação de materiais didáticos, a ser utilizado em AVA (Ambientes Virtuais de Aprendizagem), de forma a sugerir, aos usuários, materiais didáticos compatíveis com seus perfis.

Cazella et al. (2009) apresenta um modelo de sistema de recomendação de objetos de aprendizagem baseado em Filtragem Colaborativa e competências e implementa um protótipo. Ao final apresenta alguns experimentos realizados e os resultados promissores obtidos com a aplicação do modelo proposto.

Martins et al. (2011) apresenta um sistema de recomendação que tem como objetivo principal a recomendação de artigos científicos, tendo como base informações obtidas do currículo Lattes do usuário e avaliações dos usuários sobre um conjunto de artigos.

Hinz et al. (2011) propoê o uso de ontologias na modelagem do perfil do aluno em um ambiente *e-learning*, a fim de padronizar as informações e facilitar o processo de recomendação de objetos de aprendizagem

A seguir é apresentado um quadro comparativo (Tabela 10) dos trabalhos referenciados anteriomente.

Tabela 10: Síntese dos Trabalhos Relacionados

Propostas e Autores	**Tecnologias e Recursos**				**Outras Características**	**Vantagem**
	Questionário	**Ontologias**	**Sistema Multi-agentes**	**Métodos de filtragem**		
MobiLE (JACOME JUNIOR; MENDES NETO; SILVA, 2012)	Não	Sim	Sim	Contexto	Aplicação de algoritmo genético	Portabilidade para dispositivos móveis e sensíveis ao contexto
Modelo para recomendação de OA (PRIMO; VICARI; SILVA, 2010)	Não	Considera a possibilidade	Não	Híbrida	Usa padrão OBAA, considera um modelo de usuários e XML	Considera todas as principais filtragens
Infrastutura MILOS (GLUZ; VICARI, 2010)	Não	Sim	Sim	Baseada em Conteudos	Usa o padrão OBAA	Uso de agentes e ontologia

Agente de recomendação de OA (ZAÍNE, 2002)	Sim	Não	Sim	Mineração de Dados / Colaborativa	Mineração de Dados	O agente analisa recursos on-line e conteudos da *Web*
Hybrid-based Recommender System (Zhuhadar et al. 2009)	Não	Sim	Não	Híbrida e contexto	Compara com as áreas dos conteúdos educacionais	Considera uma ontologia de contexto e domínio
Arquitetura *CIECoF* (GARCÍA et al. 2009)	Não	Não	Não	Mineração de Dados / Colaborativa	Métodos de filtragem colaborativa	Utiliza de Regras de Associação
Modelo de Recomendações (FERRO, 2011)	Não	Não	Não	Híbrida	Métodos de filtragem híbrida	Criado um protótipo para a plataforma Moodle
Filtragem Colaborativa e Competências (CAZELLA et al. 2009)	Avaliação dos OA	Não	Não	Colaborativa	Foi criado um protótipo e avaliado	Considera as Competências
Sistema de Recomendação Híbrido para Bibliotecas Digitais (MARTINS et al. 2011)	Não	Não	Não	Híbrida	Métodos de filtragem colaborativa e híbrida	Extrai informações do currículo Lattes do usuário
AdaptWeb (HINZ et al. 2011)	Não	Sim	Não	Considera as possibilidades	Modelo de domínio em XML	Focado nas Ontologias

5.2 PROPOSTAS DE MODELOS DE DADOS DE ALUNOS

5.2.1 PROPOSTA DE CHEN E MIZOGUCHI

No trabalho de Chen e Mizoguchi (1999) é apresentada uma ontologia de aluno, que é composta de informações estáticas e dinâmicas. A ontologia é usada como conhecimento padrão para a comunicação entre os agentes inseridos em um ambiente multi-agente. Um dos agentes, pertencente a este ambiente, é o agente responsável pelo modelo de aluno, que troca os dados sobre o aluno com outros agentes. O agente constrói uma mensagem, que será trocada entre os sistemas, e envia juntamente a ontologia que interpreta a informação.

Um dos problemas nesse tipo de abordagem é o protocolo utilizado, pois Chen e Mizoguchi sugerem o uso de KQML (*Knowledge Query Manipulation Language)*, o que limita a abrangência dos sistemas que podem compartilhar dados. Outro problema está em definir a conceitualização e linguagem em que ambas as aplicações e o sistema de modelagem do aluno podem compartilhar seus dados. A Figura 24 apresenta a ontologia de Chen e Mizoguchi (1999).

Os dados pessoais do aluno, certificados e cursos realizados são considerados informações estáticas, porém o que se refere à informação obtida da interação entre o aluno e o sistema é considerado informação dinâmica. Podemos destacar alguns itens como objetivos, dados dos alunos e domínio de conhecimento.

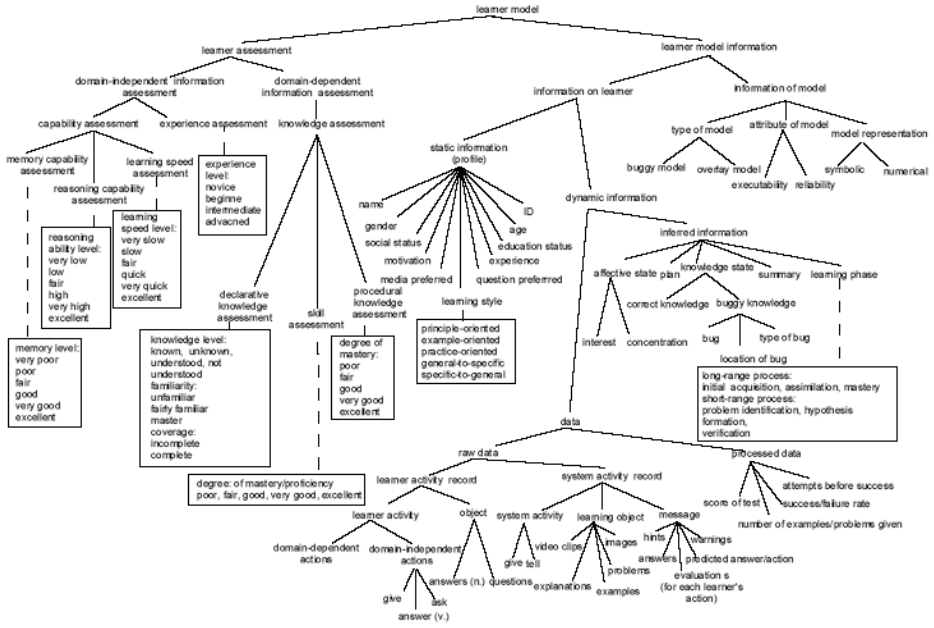


Figura 24: Ontologia do modelo de aluno proposta por Chen e Mizoguchi (1999)

5.2.2 PROPOSTA DE BRUSILOVSKY E PEYLO

Brusilovsky e Peylo (2003) apresentam um *framework* para compartilhamento de dados de alunos. O principal componente desse *framework* é um servidor de ontologias que é responsável por armazenar os modelos de aluno. O servidor de ontologias não desempenha qualquer modelagem do perfil do aluno, simplesmente armazena o seu nível de conhecimento para cada conceito da ontologia de domínio, adquirido por qualquer sistema que possua modelo de aluno. Os valores enviados por diferentes sistemas são armazenados separadamente no servidor.

O servidor de ontologias pode ser composto por múltiplos servidores, desde que todos tenham o mesmo domínio. Cada servidor armazena uma ontologia específica e as informações sobre os conteúdos educacionais e dados sobre o perfil do aluno, são expressos em termos dessa ontologia. Dessa forma, o conhecimento do aluno é modelado em diferentes sistemas, através de diferentes ontologias e armazenado em diferentes servidores de ontologias. Quando vários sistemas decidem compartilhar e/ou trocar modelos, eles devem selecionar a ontologia desejada na troca.

Após um sistema selecionar uma ontologia para modelar o aluno, o servidor de ontologias sabe o formato que deve enviar os dados sobre o aluno para esse sistema, assim como sabe onde e como requisitar dados sobre o aluno. No servidor de ontologia dois tipos de comunicação são possíveis: *push* e *pop*. Na comunicação do tipo *push*, os sistemas que

possuem dados sobre alunos enviam seus valores para o servidor e também podem solicitar valores armazenados no servidor. Na comunicação *pop* os sistemas de modelos de aluno solicitam a informação ao servidor de ontologias com algum nível de precisão. Caso os valores solicitados não estejam atualizados, o servidor buscará os valores e posteriormente enviará os dados solicitados ao seu destino. A Figura 25 apresenta a ontologia proposta por Brusilovsky e Peylo (2003).

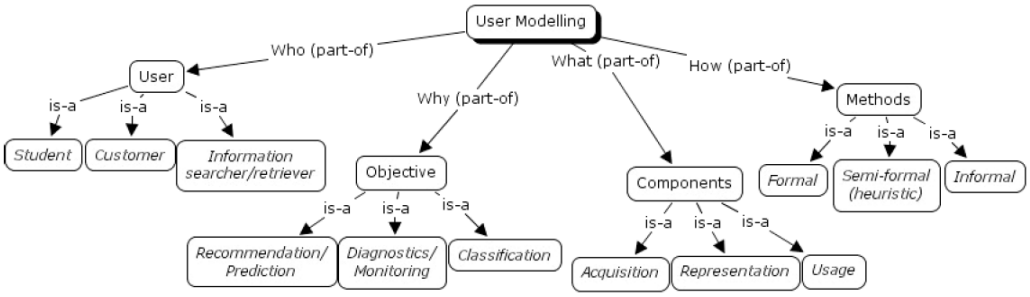


Figura 25: Ontologia proposta por Brusilovsky e Peylo (2003)

5.2.3 PROPOSTA DE NIEDERÉE

Niederée et al. (2005) propõem o uso de um modelo baseado em ontologia para a troca de dados de usuários entre sistemas. O modelo foi chamado de UUCM (*Unified User Context Model*), ou seja, modelo unificado de contexto e usuário baseado em ontologia e é a base para o processo de personalização. O modelo possui como vantagem o fato de permitir a construção de *engines* alteráveis de personalização de usuários, o que facilita a construção de ferramentas que permitem o usuário analisar e modificar a informação armazenada sobre ele. A Figura 26 apresenta o modelo UUCM.

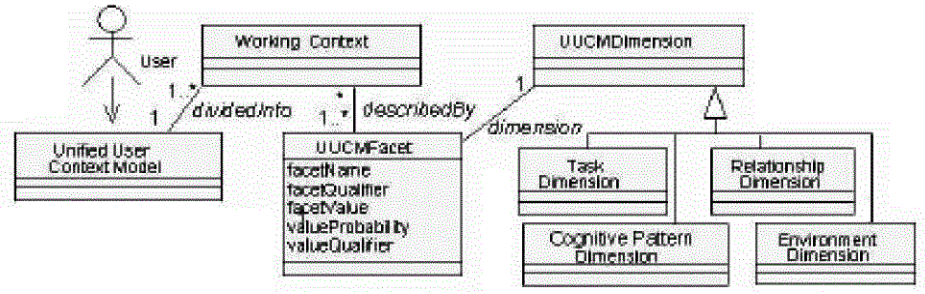


Figura 26: Modelo UUCM (NIEDERÉE et al. 2005)

O modelo UUCM está focado no contexto do usuário, que é estruturado em diferentes dimensões e considera o fato de que o usuário interage com o sistema em diferentes contextos de trabalho, estruturando o modelo de acordo com o contexto. Para permitir a personalização,

o sistema deve ser flexível e extensível o suficiente para lidar com as variações de abordagens de personalização e para incorporar os vários aspectos relevantes na captura das características do usuário e sua situação atual.

5.2.4 PROPOSTA DE DOLOG E NEJDL

Dolog e Nejdl (2007) apresentam a contextualização e implementação de um *framework* que oferece uma base comum para a troca de dados de modelos de alunos entre diversas fontes de dados. A Figura 27 apresenta a ontologia proposta por Dolog e Nejdl (2007).

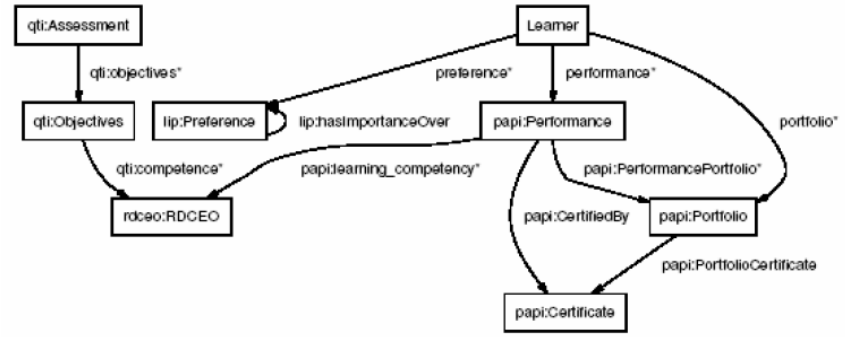


Figura 27: Ontologia para perfil de aluno proposta por Dolog e Nejdl (2007)

A ontologia é focada em dados sobre o desempenho do aluno e suas competências. Essas competências são avaliadas através dos testes realizados pelo aluno. No momento em que ele alcança alguma competência, o seu desempenho é alterado. A ontologia também compreende dados relacionados sobre as preferências do aluno como: idioma, dispositivos, recursos e preferências de estilo de aprendizagem.

Sistemas que desejam trocar dados devem utilizar a ontologia como modelo de troca. Se os sistemas possuírem dados representados em um plano diferente do modelo de troca, eles devem oferecer o mapeamento entre o seu modelo e o modelo de troca.

A ontologia proposta por Dolog e Nejdl (2007) é baseada nos padrões PAPI (PAPI, 2001) e LIP (IMS LIP, 2007) e estão centrados nos dados sobre o desempenho do aluno, seus objetivos e avaliações. Dolog e Nejdl (2007) ainda propõem um *framework* que faz uso da ontologia para tornar os perfis de alunos interoperáveis.

5.2.5 PROPOSTA DE SUDHANA ET AL.

Sudhana et al. (2013) descrevem uma proposta de recomendação de recursos educacionais que considera elementos do contexto do aluno fazendo uso de ontologias para representar este

contexto.

As informações contextuais do aluno, consideradas em Sudhana et al. (2013), estão divididas em três categorias, conforme apresentado na Figura 28: situação do aluno, domínio educacional e atividades.

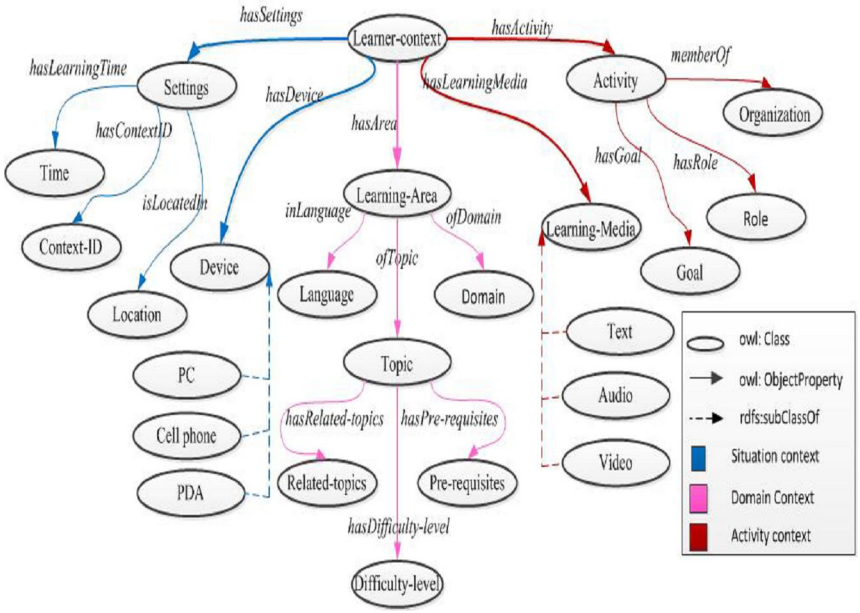


Figura 28: Ontologia parcial representando as três principais categorias de informações contextuais (SUDHANA et al. 2013)

As informações contextuais usadas são divididas como sendo parte do contexto estático ou do contexto dinâmico. No contexto estático são consideradas as informações pessoais (ex. nome, qualificação, nacionalidade) e as informações referentes ao ambiente educacional do aluno, que geralmente não estão sujeitas a constantes mudanças. Já no contexto dinâmico, são relevantes as informações relacionadas à preferência e ao interesse do aluno, que podem mudar durante o processo educacional.

A arquitetura proposta por Sudhana et al. (2013) é composta por três bases de dados: base do perfil do aluno, base de atividades educacionais e base de conhecimento. O conteúdo da base de dados do perfil do aluno é coletado através do seu registro que contém dados pessoais como qualificação, posição, preferências, entre outros. O calendário e as atividades educacionais do aluno são armazenados na base de dados de atividades educacionais. A base de dados de conhecimento armazena informações contextuais, coletadas de diferentes fontes, essas são representadas através de ontologias. Uma visão geral da arquitetura é apresentada na Figura 29.

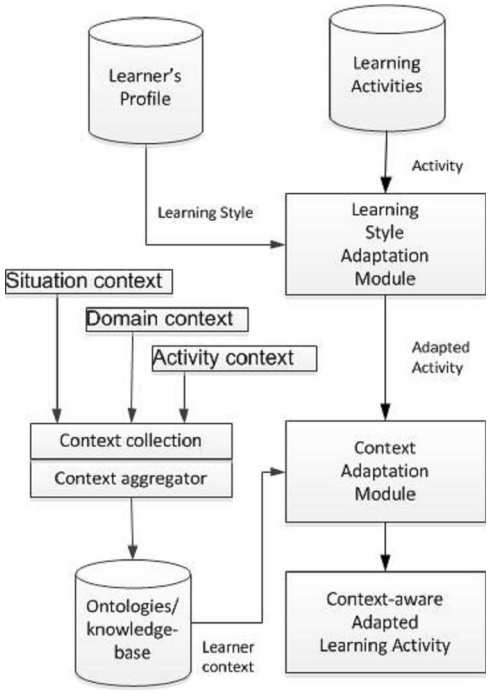


Figura 29: Visão geral da proposta arquitetural (SUDHANA et al. 2013)

5.2.6 PROPOSTA DE ALHARBI ET AL.

Alharbi et al. (2012) descrevem uma arquitetura baseada em contexto para PLEs (Recomendação em ambientes pessoais de aprendizagem). O foco da recomendação desta proposta são serviços independentes condizentes com o interesse do usuário. A recomendação é feita a partir de informações contextuais dos serviços e do aprendiz. O contexto dos serviços é adquirido através de sensores virtuais, e o contexto do usuário considera as informações do seu perfil e seu *feedback* em relação a recomendações anteriores.

As informações de contexto do usuário são categorizadas em pessoais ou estáticas e preferências. As informações pessoais consideradas nesse estudo são: nome, endereço, idade, estado civil e e-mail. O resultado da recomendação é um conjunto de serviços e recursos sugeridos a partir do interesse atual do usuário, gerando notificações de links e páginas (ALHARBI et al. 2012).

A arquitetura do PLE sensível ao contexto, apresentada na Figura 30, consiste em duas camadas: a camada do PLE e a camada provedora de serviços. A camada do PLE é composta por três módulos principais: *Personal Manager*, *Context-Aware Engine* e *User Profile*. Tais entidades se comunicam para fornecer ao usuário um serviço adaptado às suas necessidades individuais. A interface de usuário com PLE usa esta camada.

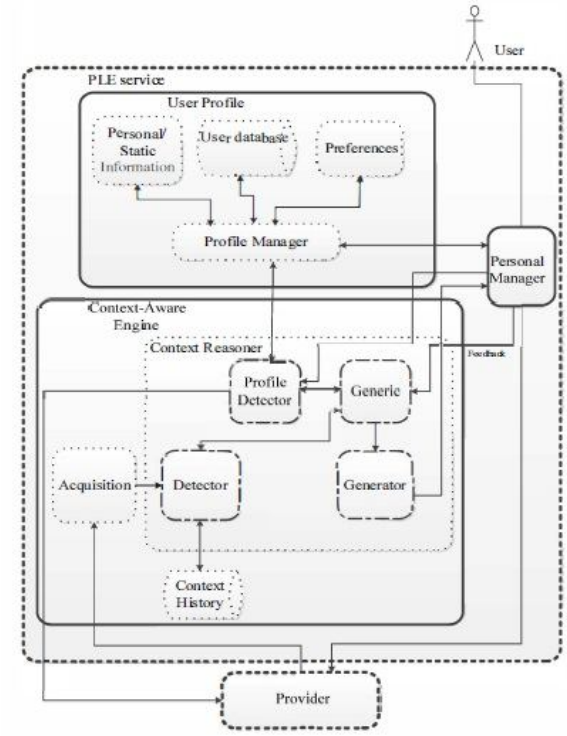


Figura 30: Arquitetura proposta por Alharbi et al. (2012)

A camada provedora de serviços é composta de várias ferramentas e prestadores de serviços independentes como, por exemplo, *Facebook*®, *YouTube*®, calendário pessoal e a biblioteca digital *IEEE Xplore*®. Ela fornece informações contextuais, através de sensores virtuais. Estas informações brutas são capturadas por APIs para interagir com a camada do PLE.

Os serviços independentes são acessados pelo usuário através do módulo *Personal Manager*. A responsabilidade de provê-los é do módulo *Context-aware Engine* que se baseia em informações contextuais coerentes com *User Profile* (ALHARBI et al. 2012).

5.2.7 PROPOSTA DE KLAŠNJA-MILICEVIC ET AL.

Klašnja-Milicevic et al. (2011) propõem uma arquitetura de um sistema de recomendação baseado em *tags* para ambientes de *e-learning*, nomeado Protus.

O Protus é um sistema de tutoria projetado para ajudar os alunos no aprendizado dos fundamentos de linguagem de programação. O principal objetivo é recomendar materiais úteis e interessantes para alunos à distância, tendo como base suas diferentes experiências, preferências, objetivos de aprendizagem e outros atributos significativos.

A arquitetura proposta por Klašnja-Milicevic et al. (2011) é representada na Figura 31.

O módulo *learner-system interaction* é responsável pela coleta dos dados sobre as atividades dos alunos (como padrões sequenciais, páginas visitadas, resultados de testes e notas obtidas) e a construção do modelo do aluno, considerando essas informações.

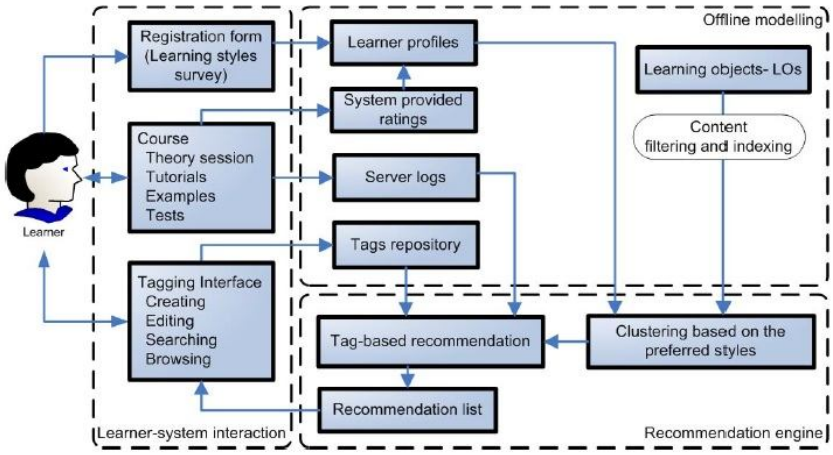


Figura 31: O componente de recomendação do sistemas Protus (KLAŠNJA-MILICEVIC et al. 2011)

O módulo *off-line* usa o modelo do aluno, construído no módulo *learner-system interaction*, para reconhecer os objetivos dos alunos, seu estilo de aprendizagem, e para filtrar objetos de aprendizagem que possuam conteúdo apropriado a eles.

O módulo *Recommendation Engine* produz uma lista de recomendação de objetos de aprendizagem. A partir dos objetos filtrados, são enviados recursos e ações recomendados, alterando a interação do aluno com o sistema. A recomendação considera ainda a capacidade do aluno em resolver uma tarefa, que pode variar em relação ao seu nível de conhecimento sobre um determinado assunto.

O Protus utiliza técnicas de clusterização para agrupar os alunos pelo seu estilo de aprendizagem. Após a definição dos grupos, eles são usados para determinar atividades educacionais apropriadas. Em seguida, uma lista de recomendação pode ser criada de acordo com *tags* de alunos e de especialistas para cada grupo gerado (KLAŠNJA-MILICEVIC et al. 2011).

5.2.8 PROPOSTA DE WANG E WU

Wang e Wu (2011) apresentam uma proposta de recomendação sensível ao contexto para implementar um sistema educacional ubíquo (*u-leaning*) adaptativo. O objetivo é permitir ao aluno construir o seu ambiente de forma auto gerenciada e personalizada.

Os autores esclarecem que a proposta está baseada nas abordagens de *inquiry-based*

learning, *social learning* e *context learning theories*. Usando as transmissões da rede sem fio e a detecção das metas de aprendizagem em situações reais através de sensores, o material didático apropriado é recomendado à plataforma de aprendizagem do aluno, considerando o seu contexto. A Figura 32 mostra o fluxo do sistema educacional ubíquo adaptativo proposto (WANG; WU, 2011).

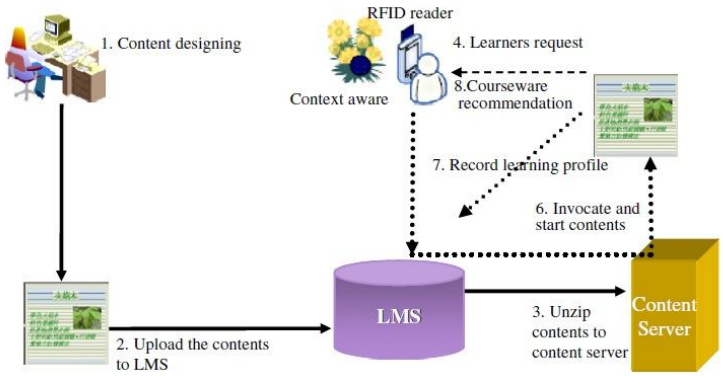


Figura 32: Sistema ubíquo adaptativo (WANG; WU, 2011)

Wang e Wu (2011) explicam o funcionamento do sistema: o aluno conecta-se à Internet através de uma rede *wireless* ou com tecnologia 3G para acessar o ambiente de aprendizado em um ambiente sem fio. Através da tecnologia RFID *(Radio Frequency Identification*), o aluno pode aplicar o leitor de RFID para detectar a localização de objetos educacionais no mundo real a qualquer momento. O sistema transmite, em seguida, os recursos de aprendizagem associados ao usuário.

O método de recomendação adotado pelo estudo utiliza a experiência dos usuários próximos que possuem metas de aprendizagens semelhantes e a avaliação feita por eles aos recursos educacionais. Após a recomendação, o aluno avalia o item recomendado e aceito por ele. Este *feedback* dado pelo aluno permite o aprendizado do sistema e uma recomendação mais qualificada, uma vez que recursos bem aceitos e avaliados são mais recomendados a outros alunos.

A definição do perfil do aluno é feita usando a técnica de Filtragem Colaborativa e Mineração dos registros de navegação na Web (*Web Usage Mining*).

5.2.9 PROPOSTA DE FRITZEN ET AL.

Em Fritzen et al. (2012) é apresentada uma proposta que busca modelar o contexto a partir da extração de mensagens trocadas entre alunos em grupos de Redes Sociais, e usar esse contexto para melhorar a relevância na recuperação de documentos na Web, tendo como

objetivo apoiar a aprendizagem colaborativa.

A intenção da proposta é tornar a recuperação de informação sensível ao contexto das discussões (por intermédio do uso das mensagens para modelagem do contexto), oferecendo, portanto, resultados de busca contextualizada.

De acordo com Fritzen et al. (2012), duas macro funcionalidades podem ser identificadas na proposta: processamento do contexto e processamento da consulta do usuário. Na Figura 33 pode-se ter uma visão geral da proposta. As setas A e B indicam as mensagens do grupo fornecendo conteúdo para o processamento do contexto, através da colaboração entre os usuários. Quando uma solicitação de consulta é identificada, conforme mostrado na seta C inicia-se o processamento para extrair os termos sugeridos para expansão e gerar uma mensagem de resposta, que terá um link para o protótipo CCS *Agent* associado à requisição gerada, indicado pela seta D. O usuário então clica no link que foi sugerido e acessa a interface do protótipo de busca, etapa essa indicada pela seta E. Ele então escolhe os termos para expandir a consulta e avalia os resultados da busca, conforme mostrado pela seta F.

Figura 33: Visão geral da arquitetura de Fritzen et al. (2012)

5.2.10 PROPOSTA DE DWIVEDI E BHARADWAJ

Dwivedi e Bharadwaj (2012) apresentam uma proposta de recomendação baseada na avaliação de conteúdo em uma rede social educacional, onde os alunos compartilham uns com os outros as suas classificações para recursos, incluindo livros, materiais didáticos, entre

outros.

Sempre que um aluno deseja uma recomendação sobre um determinado conteúdo, ele envia uma consulta para o seu amigo imediato e para os amigos distantes. O amigo imediato é aquele que está ligado a ele por um salto (ligado diretamente), enquanto que o amigo distante é aquele que está ligado a ele por 2 ou mais saltos. Os alunos que usaram o recurso no passado respondem a consulta com suas classificações. Com base nas respostas e no histórico das classificações entre amigos, é feito o cálculo da pontuação para a recomendação (DWIVEDI; BHARADWAJ, 2012).

O projeto da rede social apresentada em Dwivedi e Bharadwaj (2012) considera o estilo de aprendizagem e o nível de conhecimento de tal forma que os alunos que possuem certo grau de similaridade nesses atributos são considerado amigos imediatos. Sendo assim, a recomendação feita por um amigo imediato será compatível com o estilo de aprendizagem e o nível de conhecimento do aluno.

A proposta considera ainda a possibilidade do amigo imediato não ter tido contato com o material, cujo aluno deseja uma recomendação. Nesse caso a consulta do aluno é propagada para os amigos distantes. Além disso, a proposta explora associações de transitividade dentro da rede, por exemplo, suponha que os alunos A e B leram o livro L1 e os alunos B e C leram o livro L2, através da transitividade, é inferida a associação direta entre os alunos A e C, que passam a serem considerados amigos imediatos, conforme mostrado na Figura 34.

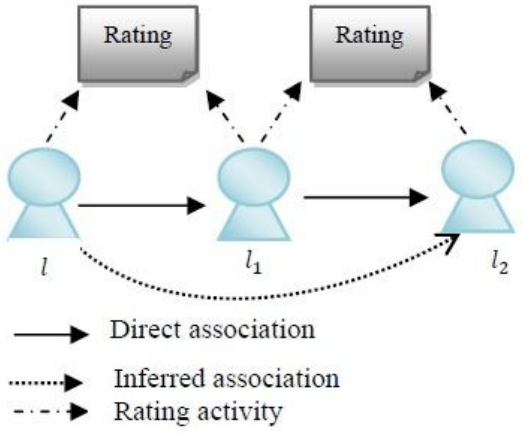


Figura 34: Associação por transitividadem (DWIVEDI; BHARADWAJ, 2012)

Um estudo experimental realizado mostrou que a precisão das recomendações baseadas em recuperação associativa, conforme esse trabalho é significativamente melhor do que a tradicional filtragem colaborativa.

5.2.11 PROPOSTA DE GASPARINI ET AL.

O *AdaptWeb*® é um sistema *e-learning* adaptativo *open source* que tem como objetivo adaptar o conteúdo, a apresentação e a navegação de cursos na web, de acordo com o modelo do usuário/aluno.

Gasparini et al. (2011) detalha uma proposta de extensão do sistema AdaptWeb® (Ambiente de Ensino-Aprendizagem Adaptativo na Web) para incorporar a noção de contexto, e para isso, foram desenvolvidos três serviços responsáveis por modelar as situações vivenciadas pelos estudantes, gerenciar e armazenar seus perfis e contextos e gerenciar seus objetos de aprendizagem correspondentes. A integração desses três serviços com os demais módulos do sistema é mostrada na Figura 35.

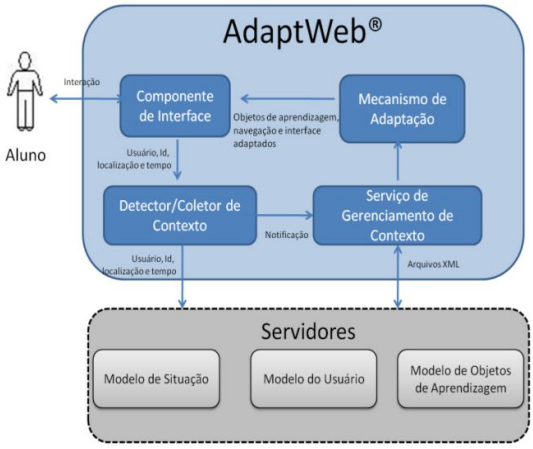


Figura 35: Arquitetura estendida do Ambiente AdaptWeb® (GASPARINI et al. 2011)

O modelo apresentado no trabalho de Gasparini et al. (2011) visa melhorar o modelo de usuário utilizando a noção de contexto. O modelo possui três níveis de abstração: meta-modelo, modelo (ontologias), e objeto. Gasparini et al. (2011) explicam que: o nível meta-modelo é representado por uma ontologia de topo (também chamada de nível superior), descrevendo conceitos abstratos como usuário, aplicativo, situação ou data, o nível do modelo expressa as diferentes dimensões contextuais, com diferentes ontologias para descrever os elementos que compõem o contexto e, no nível mais baixo - o modelo de objeto, onde são encontradas as instâncias das ontologias de contexto.

O sistema é personalizado considerando as informações disponíveis no modelo de cada aluno. São consideradas informações do contexto pessoal, tecnológico, educacional e cultural e cada dimensão é representada por uma ontologia específica.

5.2.12 PROPOSTA DE VERBERT ET AL.

Verbert et al. (2012), realizaram uma ampla pesquisa a respeito de sistemas de recomendação. Baseando-se na análise de diversos sistemas sensíveis ao contexto implantados nos últimos anos, foram definidas categorias relevantes para elementos de contexto no âmbito educacional. São elas: espaço-temporal (localização e tempo), condições físicas, computacional, recursos (virtual e física), usuário, atividades e social (Figura 36).

Os levantamentos de como os trabalhos analisados classificam as informações mostrou que não há um consenso entre os pesquisadores em relação à definição das informações contextuais e um dos desafios futuros propostos por Verbert et al. (2012) relaciona-se ao mapeamento entre as diferentes representações de dados contextuais.

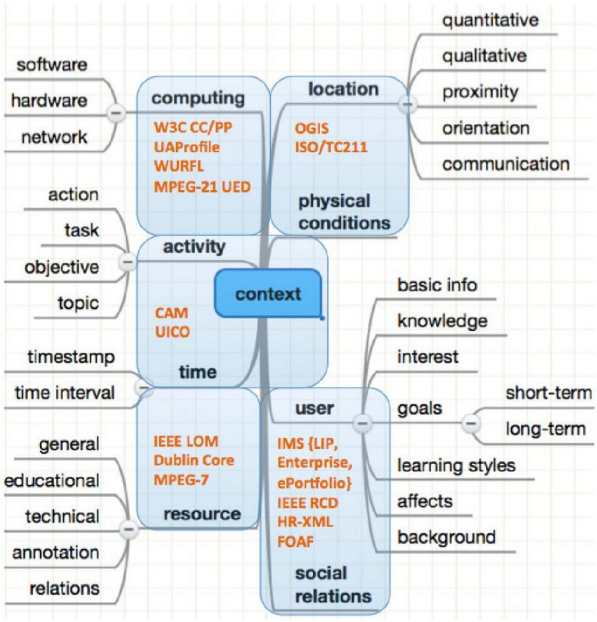


Figura 36: *Framework* de contexto (VERBERT et al. 2012)

A extração dos elementos de contexto e das técnicas de recomendação feita neste trabalho tem como objetivo identificar a relevância desses elementos na recomendação de recursos.

Analisando as propostas apresentadas (Tabela 11), percebe-se que elas possuem algumas características em comum, ou seja, consideram importante modelar informações estáticas (dados pessoais) e dinâmicas (objetivos, competências e preferências) do aluno, mas diferem em alguns pontos. Comparando as ontologias podemos dizer que as diferenças não estão na descrição de diferentes tipos de conhecimentos, mas sim na diferença de prioridades entre os tipos de conhecimento. Isso acontece em função das diferentes perspectivas em que os autores, criadores das ontologias, iniciam sua conceitualização.

Tabela 11: Elementos utilizados nos trabalhos referenciados

Propostas	Tempo	Localização	Tecnológico	Atividades	Recursos	Condições Físicas	Usuários	Relações Sociais	Educacional	Cultural
Sudhana et al. (2013)	x	x	x	x					x	
Alharbi et al. (2012)							x			
Klašnja-Milicevic (2011)				x	x		x		x	
Gasparini et al. (2011)			x				x		x	x
Dwivedi e Bharadwaj (2012)								x	x	
Fritzen et al. (2012)					x			x		
Wang e Wu (2011)			x		x			x	x	

Chen e Mizoguchi (1999) apresentam uma classificação dos termos que eles consideram importantes, os quais são usados posteriormente por agentes para troca de mensagens. As ontologias de Dolog e Nejdl (2007) e Chen e Mizoguchi (1999) possuem em comum o fato de modelarem questões pessoais do aluno, como suas preferências e objetivos. A abordagem de Niederée et al. (2005) preocupou-se em modelar as áreas de interesse do aluno, suas competências (habilidades e especialidades) e preferências dentro de um determinado contexto, pois segundo o autor, as características de um aluno podem ser modificadas de acordo com o seu contexto.

A abordagem proposta por Brusilovsky e Peylo (2003) não apresenta uma ontologia específica e sim de um servidor de ontologias, que é responsável pela troca dos dados. Nela podemos destacar alguns itens como: objetivos, métodos, componentes e dados dos alunos. Já na proposta de Niederée et al. (2005) podemos destacar alguns itens como: tarefas e relações com outros usuários.

A ontologia proposta por Dolog e Nejdl (2007) tem termos de modelagem de dados sobre o aluno, pode ser considerada a mais completa, pois considera informações pessoais dos alunos, suas avaliações, preferências, objetivos, desempenho, certificados e portfólio. Porém ela não considera informações sobre o contexto do aluno, ponto considerado na abordagem de Niederée et al. (2005). Nela também se verifica que o autor representa apenas os conceitos essenciais, relacionados com as preferências, desempenho, certificados e avaliações do aluno, e que os mesmos seguem os padrões PAPI (PAPI, 2001) e LIP (IMS LIP, 2007), mantendo uma correspondência direta com o domínio e terminologias existentes, satisfazendo os requisitos de clareza e legibilidade. Embora a presença de padrões restrinja a flexibilidade da descrição do modelo do aluno, isto torna a solução mais geral, pois a mesma pode ser aplicada em qualquer sistema que reconheça o padrão.

Entre os elementos contextuais utilizados foi possível observar que não há um consenso em relação à sua categorização, apesar de existir uma coincidência dos elementos usando uma nomenclatura distinta (VERBERT, 2012).

Foram observadas também as estratégicas empíricas usadas para avaliar e/ou validar as propostas em cada um dos trabalhos. Apenas três dos 7 trabalhos utilizaram algum tipo de estratégica para avaliar a proposta.

Detectou-se nos trabalhos o uso associado de diferentes técnicas de recomendação para obter melhores resultados (FRITZEN et al., 2012; KLAŠNJA-MILICEVIC 2011; WANG; WU, 2011), além de algumas modificações em técnicas clássicas, como a filtragem colaborativa, com o intuito de suprir as suas deficiências (DWIVEDI; BHARADWAJ, 2012).

Os autores não falam da metodologia utilizada para escolha dos dados contidos no modelo do aluno, apenas classificam dados baseados em sistemas que as implementam.

5.3 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo foi apresentada uma visão de trabalhos relacionados de sistemas sensíveis ao contexto com análises dos tipos de sistemas, tecnologias, recursos, metodos e características. Ficou evidenciado que poucas propostas utilizas, simultaneamente, recursos como ontologias de domínio, multi-agentes e uma análise sensível ao contexto do usuário.

Outra lacuna encontrada é que nenhum dos trabalhos considera a análise do professor, que neste trabalho é apresentado como plano didático. O plano didático do professor detalha os procedimentos para atingir um dado objetivo educacional. Ele identifica elementos do contexto do aluno considerando as tarefas, objetivos e comperências a serem alcançadas e seleciona os objetos de aprendizagem que irão integrar o seu plano.

Também, nessa revisão foi dada ênfase nas catacterísticas em comum dos modelos em função das diferentes perspectivas e destacamos que: a aquisição de contexto, bem como a definição dos elementos apropriados ainda são um desafio, pois muitos trabalhos utilizam elementos contextuais adquiridos através de operações manuais, como entrada de dados do usuário, apesar de alguns trabalhos já considerarem formas de aquisição automáticas. Assim, para a arquitetura BROAD-RS, foi proposta a ontologia PERSONNA, que tem por objetivo servir como base para modelagem do perfil e contexto do aluno, a fim de facilitar o processo de recomendação de objetos de aprendizagem.

6 BROAD-RS: SISTEMA PARA RECOMENDAÇÃO DE OA SENSÍVEL AO CONTEXTO

Este capítulo apresenta o histórico do projeto BROAD, a arquitetura BROAD-RS proposta por este trabalho, a estrutura do plano didático desta arquitetura, seus agentes com suas especificações, a estrutura da ontologia PERSONNA, o protótipo desenvolvido e alguns cenários de aprendizagem.

O foco desta pesquisa está na recomendação e no gerenciamento de conteúdos personalizados. A arquitetura proposta, denominada BROAD-RS (*BROAD Recomendation System*) (REZENDE et al. 2013), tem o objetivo de realizar recomendações dos OAs adaptados aos interesses do aluno com o uso de uma rede de ontologias e sistema de agentes. Este modelo utiliza os metadados do padrão BROAD (CAMPOS et al. 2011b).

O sistema multi-agentes leva em conta elementos do plano didático definidos pelo professor e o contexto e perfil do aluno para recomendar os OAs mais adequados. O contexto é o ambiente físico ou conjunto de circunstâncias (como o local e o tempo) a partir do qual se considera um fato que ajuda a compreender os elementos envolvidos. Aplicações cientes de contexto são aplicações que são capazes de modificar seu comportamento baseado nas informações de contexto ou são aplicações que mostram ao usuário informações de contexto.

As motivações de usar um sistema sensível ao contexto são que as aplicações estejam de acordo com os elementos do ambiente e poderam adaptar-se dinamicamente às mudanças e às necessidades correntes do aluno. Assim podem explorar características do ambiente, tais como: a localização do aluno, as suas características, hora do dia, estado do ambiente físico, estado do sistema computacional, histórico de interações do aluno, entre outras.

Ao conhecer as necessidades dos alunos (parte das informações contextuais) os agentes podem ser responsáveis pela coleta automática de informações para a rede de ontologias de acordo com os critérios estabelecidos pelo professor em seu plano didático. O plano didático é o local onde o professor seleciona e avalia quais os OA são considerados para a recomendação.

6.1 PROJETO BROAD

O projeto BROAD (Busca e Recuperação de Objetos de Aprendizado a Distância) busca a adoção de tecnologias como ontologias, serviços web semânticos, agentes e *workflow* para a construção de uma arquitetura para composição e sequenciamento de OA. O projeto avança

em questões relativas a trabalhos anteriores como em arquiteturas como SASAgent (MENDES et al., 2011), BROAD (BRAGA et al., 2011) (CAMPOS et al., 2012a), em repositórios semânticos como (SANTOS et al., 2008b) e objetos de aprendizagem (CAMPOS et al., 2011b) (SOUZA et al., 2010).

Segundo Campos et al. (2011), a reutilização de objetos de aprendizagem mostra seu potencial para acelerar a preparação e a composição de cursos à distância, e assim contribuir para aumentar a oferta de educação. Um aspecto que pode prejudicar a disseminação do uso de objetos de aprendizagem é a dificuldade de localizar e recuperá-los de acordo com as necessidades dos usuários finais. Para facilitar a localização e recuperação desses objetos de aprendizagem tem-se que: o OA deve ser registrado em conformidade com um modelo claro e aplicável de metadados, os artefatos devem ser armazenados em repositórios Web e as buscas nesses repositórios devem ser guiadas por termos semanticamente ricos, claros e não ambíguos.

No projeto BROAD (NERY et al. 2011) e (CAMPOS et al. 2011b) foi desenvolvido uma ontologia de representação de objetos de aprendizagem e seus metadados, mostrada na Figura 37.

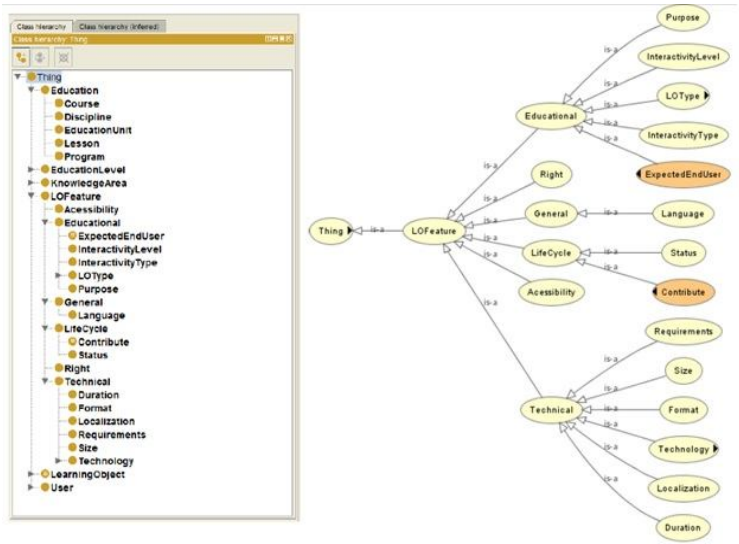


Figura 37: Ontologia BROAD (NERY et al. 2011)

As ontologias vêm sendo apontadas como solução para a semântica de aplicações computacionais na área de educação, prioritariamente com OAs, potencializando seu uso a partir de repositórios, na busca e cadastro, e também na composição destes OAs para a formação de conteúdos educacionais. A rede de ontologias BROADNet de objetos de aprendizagem, é usada no suporte semântico a aplicações do projeto (CAMPOS et al. 2012b).

A ligação de diferentes ontologias em rede, através de técnicas que garantam a

interoperabilidade entre elas pode significar a inclusão de conteúdos já existentes ou complementares para ampliação da semântica nas aplicações computacionais (Figura 38).

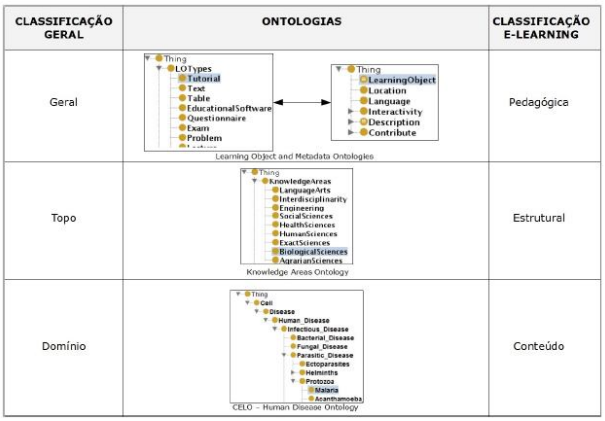


Figura 38: Classificação das principais ontologias da rede BROADNet (CAMPOS et al. 2012b)

A infraestrutura BROAD-WP, (NERY et al. 2011), propõe a especificação de uma arquitetura para geração de *workflow* de conteúdos educacionais de forma a personalizar a oferta conteúdos educacionais focada no uso de *workflow* e não contextualiza a recomendação.

Segundo Nery et al. (2011), a infraestrutura BROAD-WP é uma extensão a infraestrutura BROAD através da utilização de diferentes objetos de aprendizagem de acordo com as características do aluno. Segundo Campos et al. (2012b), com o objetivo de potencializar o reúso de OA uma infraestrutura, denominada BROAD-PL, (Figura 39), foi projetada buscando-se estender a infraestrutura BROAD-WP através da abordagem de LPS (Linha de Produtos de Software) e da utilização de uma arquitetura orientada a serviços. Considerando-se os mecanismos que apoiam a construção de OA e que facilitam a integração com Ambientes Virtuais de Aprendizagem, a abordagem de LPS pode oferecer um suporte sistemático para a reutilização a partir das características e preferências do aluno.

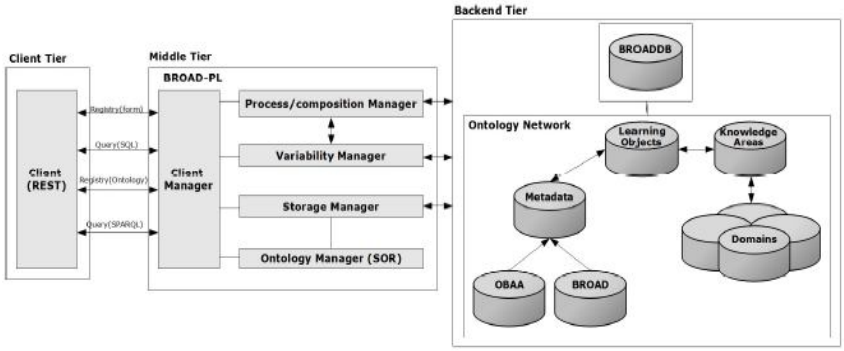


Figura 39: Arquitetura Proposta para o Projeto BROAD-PL (CAMPOS et al. 2012b)

6.2 ARQUITETURA BROAD-RS

No projeto BROAD, realizar recomendações de objetos de aprendizagem através de contexto, metadados e planos didáticos é uma evolução natural da pesquisa.

Pensando nisso, o presente trabalho apresenta uma arquitetura (Figura 40) que, através de recomendação qualificada dos OA feita pelo professor a partir do seu plano didático e de um sistema multi-agentes, realiza recomendações de objetos educacionais (OAs) que sejam adequados às características do contexto do estudante.

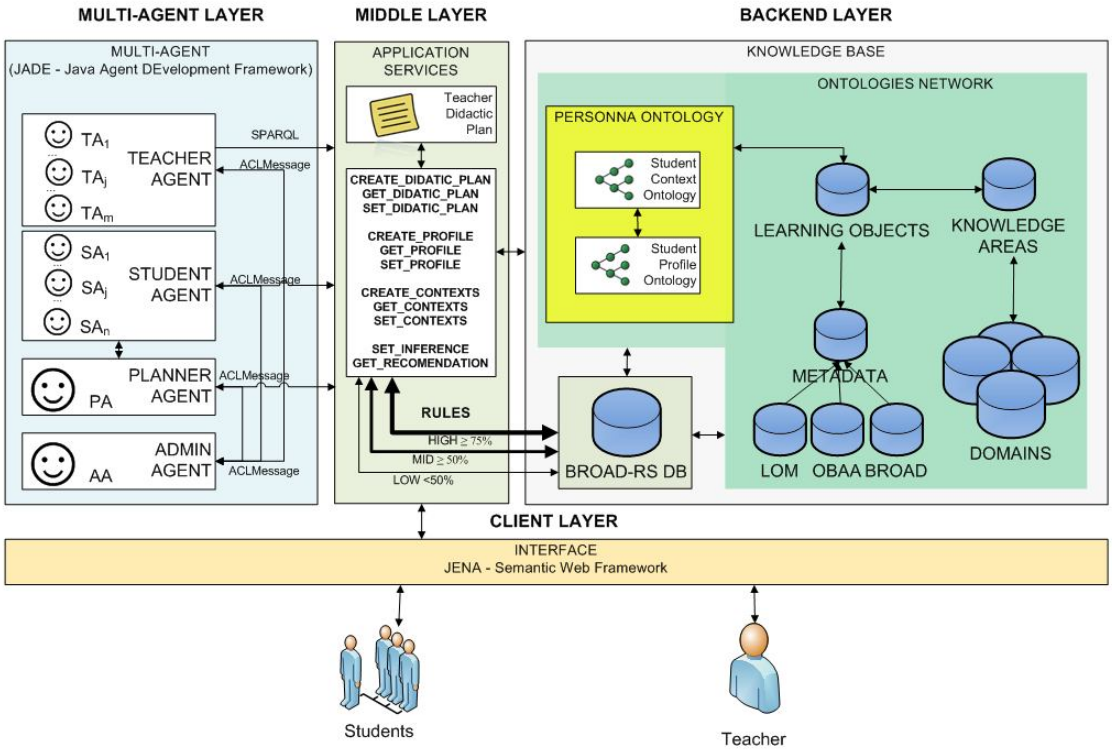


Figura 40: Arquitetura BROAD-RS

A arquitetura de recomendação de objetos de aprendizagem proposta permite a automação do processo conforme perfil e contexto do aluno e de acordo com o plano didático especificado pelo professor. Para auxiliar na recomendação de objetos de aprendizagem é utilizada a rede de ontologias.

A arquitetura está estruturada em quatro níveis:

- *Multi-agent layer* (Nível de Agentes): esta camada descreve todos os agentes, seus respectivos papeis e suas inter-relações e interações com as aplicações. Os agentes relacionam os dados de contexto e perfil do aluno e plano didático do professor encontrados no banco de dados com a rede de ontologias. Como esta plataforma deve prover escalabilidade e prover recursos para a comunicação,

os agentes que compõem a arquitetura estão baseados em três características básicas: reatividade (por reagir a estímulos/mensagens recebidas de outros agentes), pró-atividade (ações sem um evento, por exemplo, o agente *admin* monitorar a disponibilidade dos agentes) e, finalmente, a habilidade socializar (capacidade do agente para interagir com outros agentes).

- *Middle layer* (Nível Médio): é a camada de serviços, utilizada para acessar as aplicações da arquitetura. É responsável pela interação dos agentes com a base de conhecimento. É nesta camada que se encontram serviços de criação, gravação e recuperação de dados de contexto, perfil e plano didático, e serviços de realização das maquinas de inferências das ontologias.
- *Backend layer* (Nível Baixo): é a camada de conhecimento, onde se encontra a rede de ontologias e onde os OAs são armazenados. É nela que se encontra a *Personna Ontology (ontologia de contexto e perfil de aluno)*, ontologias de áreas do conhecimento e domínio *e a Learning Object Ontology*. A *Learning Object Ontology* caracteriza os recursos usando o padrão de metadados BROAD (CAMPOS et al. 2011b). É nesta camada que a aplicação armazena os planos didáticos e as recomendações já realizadas em um banco de dados (BROAD-RS DB), interagindo diretamente com a base de conhecimento, ampliando a capacidade de expressão semântica das informações de contexto e perfil do aluno.
- *Client layer* (Nível de Cliente): implementa o cadastro de objetos de aprendizagem, o cadastro de alunos e professores e a interface e telas dos usuários que pode ser desenvolvida em qualquer linguagem ou *framework*.

6.3 PLANO DIDÁTICO DO PROFESSOR

Uma das carcteristicas desta arquitetura é o suporte oferecido para estabelecer o local onde o professor define seus planos didáticos pedagógicos.

O plano didático do professor detalha os procedimentos para atingir um dado objetivo educacional. Ele identifica elementos do contexto do aluno considerando as tarefas e objetivos e competências a serem alcançadas.

O plano didático na arquitetura BROAD-RS é um mecanismo de recomendação baseado em um conjunto de metadados que permitirão o professor avaliar um OA e através dessa avaliação identificar a relevância desse objeto para o plano didático planejado para o

aluno e gerar, então, uma recomendação qualificada para esse OA.

Para a definição do plano didático da arquitetura BROAD-RS foram considerados os elementos de três padrões de metadados educacionais, os padrões BROAD (CAMPOS et al. 2011b), LOM (IEEE, 2012) e o OBAA (VICARI et al. 2009). Os metadados educacionais considerados do padrão LOM e BROAD para o plano didático são: *Learning Resource Type, Interactivity Type, Interactivity Level, Semantic Density, Context, Typical Age Range, Difficulty, Language*. Os metadados educacionais selecionados do padrão OBAA são: *Learning Content Type, Interaction e DidacticStrategy*. O metadado *Context* do padrão LOM (IEEE, 2012) foi renomado para *Learning Context*.

Na Tabela 12 apresentamos com detalhe a estrutura dos metadados do plano didático para a arquitetura BROAD-RS. Verifica-se que o *Learning Resource* depende dos dados escolhidos em *Media, Type Level, Interactivity Level e Perception*. Esta dependência é representada pelas colunas "Referência com", exemplo: o tipo de recurso *Articles* somente será disponibilidado se as mídias *Texts* ou *Audios* ou *Web* forem escolhidas e o tipo de interatividade for *Active* e o nivel de interatividade for *Very low* e a percepção for *Textual* ou *Auditory* considerados do padrão LOM.

Tabela 12: Metadados do Plano Didático da Arquitetura BROAD-RS

Metadados	Descrição	Opções
Language	Linguagem	[X] Portuguese
		[X] English
		[X] Others
Media	Mídia	[1] Videos
		[2] Texts
		[3] Images
		[4] Audios
		[5] Web
		[6] Software
Perception	Percepção	[A] Visual
		[B] Auditory
		[C] Textual
		[D] Tactile
Interactivity Level	Nivel de Interatividade	[VH] Very high
		[H] High
		[M] Medium
		[L] Low
		[VL] Very low
Interactivity Type	Tipo de Interatividade	[I] *Active*
		[II] *Expositive*
		[III] *Mixed*

		Descrição	Referência com *Media*	Referência com *Type Level*	Referência com *Interactivity Level*	Referência com *Perception*
Learning Resource	Tipo de Recurso	*[X] Articles*	(2,4,5)	I	VL	CB
		[X] Books	(2,4,5)	II	VL	CB
		[X] Record Voices	(1,4,5)	II	L	B
		[X] Music	(1,4,5)	II	L	B
		[X] Movies	(1,5)	II	L	ABC
		[X] Maps	(3,5)	II	M	AC
		[X] Sites	(5)	III	VH	ABC
		[X] Models	(3,5)	II	L	ACD
		[X] Graphs	(1,2,5)	II	L	AC
		[X] Simulations	(1,3,4,5,6)	I	H	ABC
		[X] Texts	(2,5)	II	VL	C
		[X] Questionnaire	(2,5)	I	H	C
		[X] Photos	(3,5)	II	VL	A
		[X] Diagrams	(2,3,5)	II	M	AC
		[X] Animation	(1,3,5,6)	II	M	ABC
		[X] Magazines	(2,3,5)	II	L	C
		[X] Presentation	(1,2,3,4,5,6)	II	M	ABC
		[X] Application	(5,6)	III	VH	ABC
		[X] Manuals	(1,2,3,4,5,6)	II	M	ABC
		[X] Exercises	(1,4,5,6)	I	M	ABC
		[X] Spreadsheets	(2,5)	I	H	C
		[X] Tutorials	(1,2,3,4,5,6)	II	M	ABC
		[X] Game	(5, 6)	III	VH	ABCD
Difficulty	Dificuldade	*[X] Very high*				
		[X] High				
		[X] Medium				
		[X] Low				
		[X] Very low				
Age Range	Faixa Etária	*[7] Age Min / [9] Age Max*				
Learning Context	Escolaridade	[X] *Higher Education* - Educação Superior				
		[X] *Professional Education* -Educação Profissional				
		[X] *School* - Ensino Médio				
		[X] *Elementary Education* - Ensino Fundamental				
		[X] *Childhood Education* - Ensino Infantil				

O metadado *Semantic Density* do padrão LOM, e do padrão OBAA, os metadados *LearningContentType* e *Didactic Strategy* não serão considerados para este plano didático, e sim, para trabalhos futuros devido à complexidade de implementação e análise.

6.4 AGENTES DA ARQUITETURA

A relação entre os alunos e a arquitetura se faz com a utilização de agentes, que recomendam objetos de aprendizagem com a utilização dos serviços da camada *Middle layer*. Esta abordagem facilita a modularidade, a flexibilidade e a escalabilidade do ambiente de

aprendizagem.

A organização do SMA (Sistema Multi-Agente) é do tipo comunidade de especialistas, pois cada um dos quatro tipos de agentes criados encontra-se no mesmo nível, sendo que cada um deles é especialista em determinada tarefa. Os agentes interagem entre si através de um protocolo de comunicação ACL (*Agent Communication Language*) previamente estabelecido. No momento em que um aluno acessa o BROAD-RS os agentes, analisando o seu contexto e os planos didáticos construídos pelos professores, recomendam objetos de aprendizagem com o uso das ontologias.

A seguir é descrito cada um dos agentes:

- Agente do professor (TA - *TEACHER AGENT*): é responsável por todas as atividades desenvolvidas pelo usuário professor no ambiente e cada professor é instanciado. O TA é responsável, principalmente, por verificar se houve mudanças nas informações cadastradas pelo professor, como alterações em seus planos didáticos.

- Agente do estudante (SA - *STUDENT AGENT*): tem como principal objetivo identificar necessidades dos estudantes, tais como: identificar os conteúdos de maior interesse, as suas mídias prediletas, considerar seu nível de aprendizagem, entre outras. Para que o SA obtenha sucesso nos seus objetivos ele conta com uma base de dados de recomendações já realizadas e a ontologia PERSONNA estruturando as informações do estudante que serão tomadas como referencial para a busca de objetos de aprendizagem mais compatíveis ao contexto e perfil. Assim como o TA, o SA representa o estudante no ambiente e cada estudante tem o seu SA. Ou seja, os SA são responsáveis por monitorar as atividades dos estudantes, atualizar e recuperar suas preferências e interagir com a ontologia e o banco de dados.

- Agente de planejamento (PA - *PLANNER AGENT*): indica para aplicação quais são os objetos de aprendizagem a serem apresentados aos alunos. É ele que recebe as informações dos outros agentes e avalia se é necessário chamar a máquina de inferência das ontologias em rede. Caso não houver comunicação dos outros agentes o PA recomenda os OA baseado em recomendações antigas armazendas no banco de dados relacional. Após identificar os OAs adequados, o PA armazena as informações das recomendações no banco de dados para essas consultas posteriores.

- Agente administrativo (AA - *ADMINISTRATION AGENT*): é responsável por controlar os demais agentes. Ele monitora o ciclo de vida de todos os outros agentes, seus serviços e também as suas interações com o ambiente.

6.4.1 PROJETO E ESPECIFICAÇÃO DOS AGENTES

A arquitetura dos agentes é projetada a partir das diretivas da metodologia INGENIAS, (GÔMES-SANZ; FUENTES, 2009), que representa os requisitos do sistema e considera o uso de seis modelos: diagramas de casos de uso (do mais geral para os mais específicos), modelo de organização (visão da estrutura geral do SMA incluindo os papéis dos agentes, relações das características de cada um e o fluxo de trabalho), modelo de agente (representando as tarefas que os agentes devem realizar para alcançar seus objetivos), modelo de objetivos/tarefas (identifica todos os objetivos, tarefas gerais e específicas existentes na arquitetura que podem ser associadas aos agentes), modelo de interação (representando a interação dos agentes com seus respectivos papéis) e um modelo do ambiente externo (especificando as entidades externas existentes e como se relacionam com o SMA).

Detalhamos a seguir os modelos resultantes da utilização da metodologia INGENIAS para a modelagem da arquitetura BROAD-RS.

6.4.1.1 DIAGRAMAS DE CASO DE USO

Os diagramas de casos de uso da arquitetura proposta foram divididos em duas categorias: gerais e específicos. Para os diagramas específicos é importante salientar que os usuários se dividem em dois grupos distintos: professores e alunos. Estes dois grupos podem ser entendidos como papéis que cada usuário pode assumir dentro da plataforma.

Alguns diagramas de caso de uso são:

- geral: os atores são professores e alunos, e engloba o gerenciamento, utilização e recomendação dos OA (Figura 41);
- específico 1: são recomendados aos alunos OAs que atendem aos seus contextos e perfis através de inferência da rede de ontologia que é gerenciada pelos agentes (Figura 42);
- especifico 2: para o sistema de recomendação, os professores criam seus planos didáticos avaliando os objetos de aprendizagem (Figura 43).

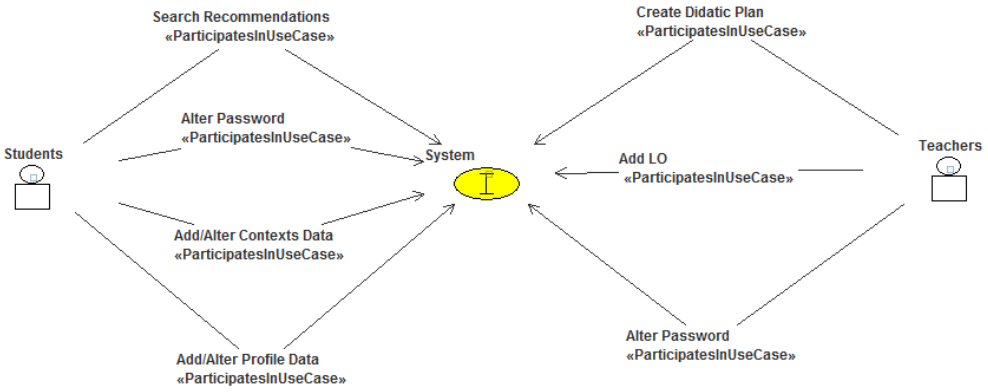


Figura 41: Diagrama de Casos de Uso: Geral

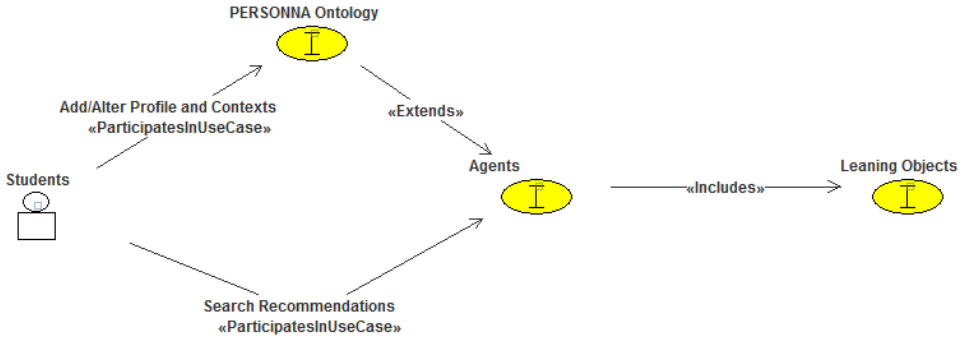


Figura 42: Diagrama de Casos de Uso: Específico 1

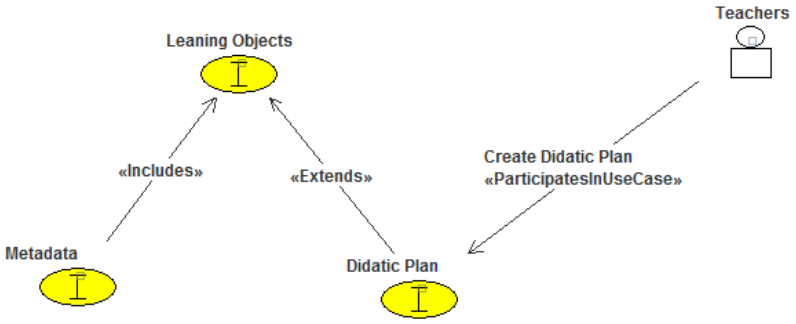


Figura 43: Diagrama de Casos de Uso: Específico 2

6.4.1.2 MODELOS DE ORGANIZAÇÃO

O diagrama de organização da metodologia INGENIAS é responsável por representar os diferentes componentes do sistema (agentes, papéis, recursos e aplicações), a funcionalidade do sistema e possíveis restrições que porventura existam sobre as interações entre os agentes.

Na arquitetura proposta neste trabalho, como representação da organização dos compontes do sistema, foram identificados três grupos (*Framework_Group, Student_Group e Teacher_Group*) que precisam ser representados, ambos a partir de uma mesma origem:

System_Group ou seja, grupo de todo o sistema (Figura 44). Assim, usuários possuem dois papéis principais dentro da arquitetura, podem ser estudantes (*Students*) ou podem ser professores (*Teachers*).

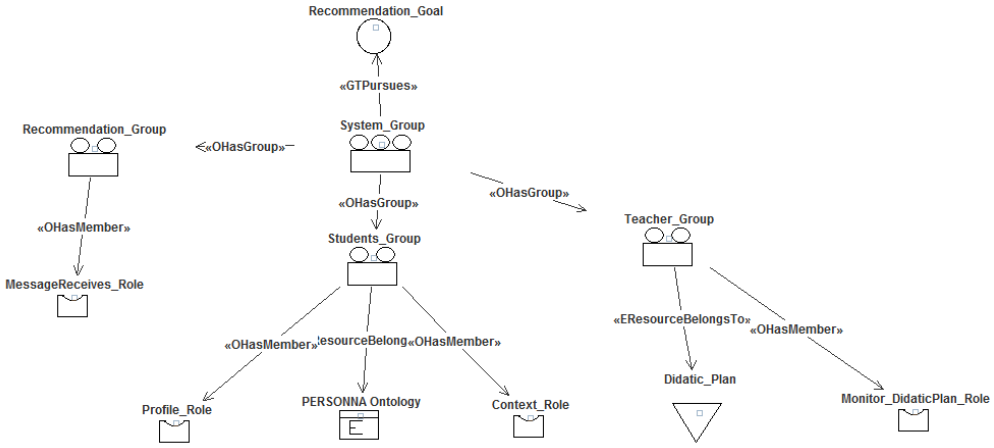


Figura 44: Modelo de Organização – Geral

Para que o objetivo da arquitetura seja alcançado temos papéis que os agentes assumem durante todo o processo. O *MessageReceives_Role* tem como objetivo receber informações dos outros agentes para realizar a recomendação de objetos de aprendizagem. Os papeis *Profile Role, Context Role e Monitor_Didatic_Plan* tem como objetivo atender às necessidades dos alunos e professores, ou seja, monitorar se o professor alterou seu plano didático ou se o aluno alterou seu pefil e contexto.

É importante salientar que existe o grupo que representa os agentes do *framework* de suporte a implementação da arquitetura (Figura 45) (agentes nativos da plataforma JADE). Estes agentes são os responsáveis pelo gerenciamento de toda a arquitetura e ainda responsáveis em fazer a interação entre os usuários finais (professor ou aluno) através de uma interface amigável com o BROAD-RS.

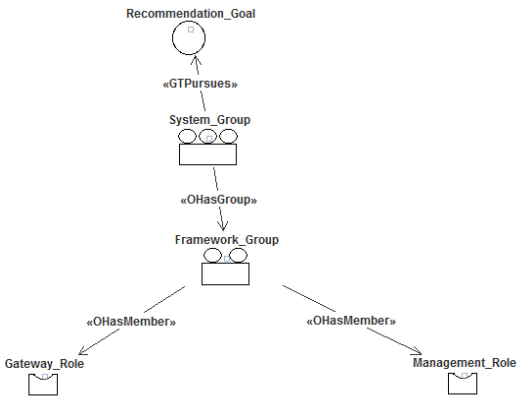


Figura 45: Modelo de Organização – Específico 1

6.4.1.3 MODELOS DE AGENTES

O modelo de agentes prevê a representação dos agentes mais importantes da arquitetura de maneira particular. Seus papéis e seus objetivos desempenhados dentro da arquitetura são o foco destes modelos. Na arquitetura proposta podemos destacar quatro agentes pela importância individual: *Student Agent*, *Planner Agent*, *Teacher Agent* e *Admin Agent.*

O *Student Agent* (Figura 46) é o agente responsável por representar o aluno na arquitetura. Este agente assume dois papéis: monitorar os dados do contexto e perfil do aluno e enviar mensagem para o *Planner Agent* quando ocorrer essas alterações.

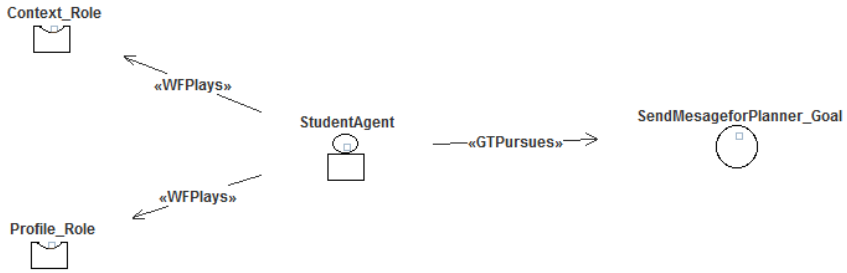


Figura 46: Modelo de Agentes – *Student Agent*

Já o *Planner Agent* recebe informações dos outros agentes. Seu objetivo principal é executar a recomendação de objetos de aprendizagem, ele também salva informações no banco de dados e executa a máquina de inferência das ontologias (Figura 47).

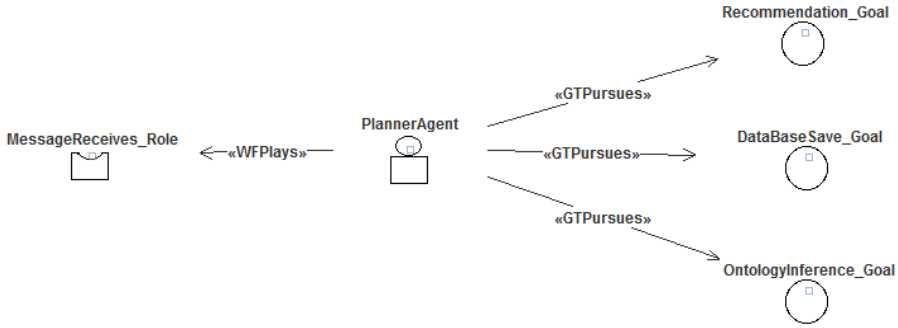


Figura 47: Modelo de Agentes – *Planner Agent*

Na Figura 48 temos a representação do *Teacher Agent*, é o agente responsável por representar o professor na arquitetura. Este agente monitora alterações dos dados do plano didático do professor e tem como objetico enviar mensagem para o *Planner Agent* quando ocorrerem essas alterações.

Figura 48: Modelo de Agentes – *Teacher Agent*

Finalmente temos o *Admin Agent* (Figura 49), responsável pelo monitoramento de todos os outros agentes e serviços da arquitetura.

Figura 49: Modelo de Agentes - *AdminAgent*

6.4.1.4 MODELOS DE TAREFAS E OBJETIVOS

Os diagramas ou modelos de tarefas e objetivos representam as relações entre objetivos e tarefas além das suas estruturas internas. Permitem também indicar as entradas e saídas de cada tarefa e quais são os efeitos das mesmas tanto na arquitetura, de modo geral, como no estado de cada agente.

A Figura 50 representa o modelo de tarefas e objetivos em nível macro. O objetivo da arquitetura proposta é realizar recomendação de objetos de aprendizagem. Este objetivo principal pode ser decomposto em três outros objetivos que precisam ser atendidos pela plataforma: envio de mensagem, chamada da maquina de inferência da ontologia e salvar no banco de dados.

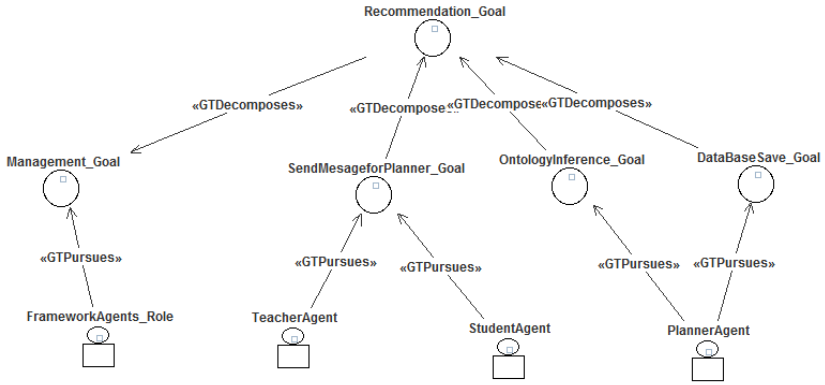


Figura 50: Modelo de Tarefas e Objetivos - Nível Macro

Na Figura 51 é apresentado o modelo de tarefas e objetivos que representam e detalham o objetivo de gerenciamento da arquitetura. Este objetivo é garantido pelo *framework* de agentes JADE (BELLIFEMINE, 2003) e tem como função principal gerenciar o ciclo de vida dos objetos de aprendizagem. Temos para isto três agentes especializados: *AMSAgent, DFAgent* e *Admin Agent.*

O *AMSAgent* é o responsável pelo registro dos agentes do *framework* JADE, semelhante ao serviço de páginas brancas das listas telefônicas. O *DFAgent* é responsável pelo serviço de páginas amarelas do *framework* JADE controlando as descrições sobre as

funcionalidades disponíveis de cada agente que está sendo executado no *framework* JADE.

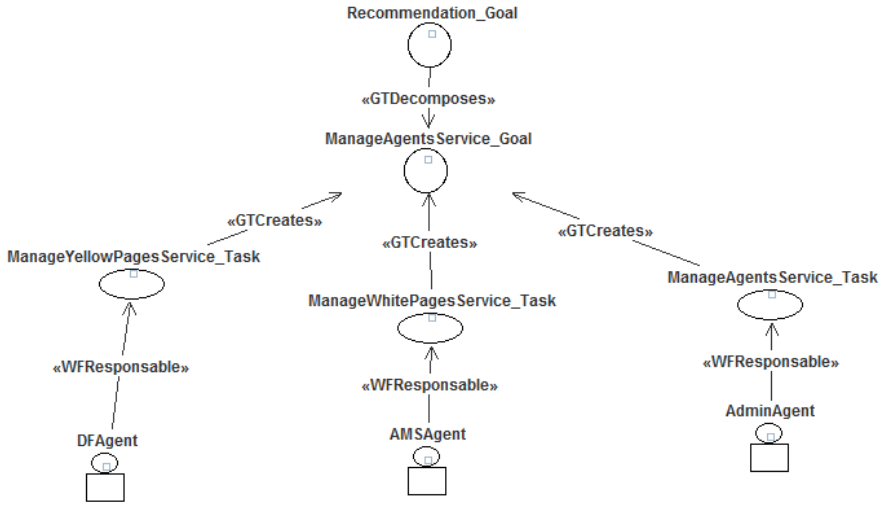


Figura 51: Modelo de Tarefas e Objetivos - Gerenciamento

O objetivo de recomendar objetos de aprendizagem proposto é permitir que o aluno receba OAs de acordo com os seus contexto e perfil. Estes critérios são recebidos pelo agente *Planner Agent.* A recomendação é feita através de um processo de inferência com a utilização de uma rede de ontologias, relacionando dados de contexto e perfil dos alunos e o plano didático do professor (Figura 52).

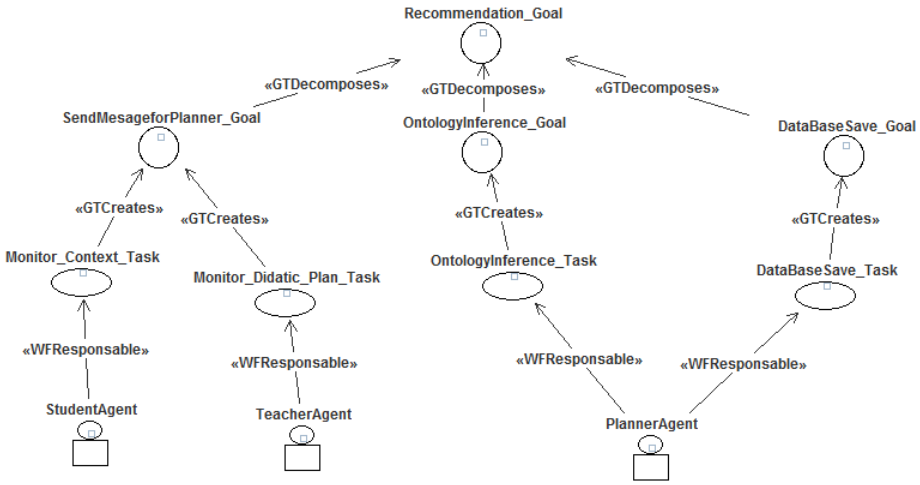


Figura 52: Modelo de Tarefas e Objetivos – Recomendação

6.4.1.5 MODELOS DE INTERAÇÃO

O modelo de interação descreve como são as interações entre os agentes na arquitetura BROAD-RS. A descrição do modelo envolve atores e os objetivos que precisam ser alcançados através das interações. Com isto, é fácil observar que o modelo de interação possui um relacionamento estrito com o modelo de tarefas e objetivos.

A partir dos modelos de tarefas e objetivos, os modelos de interação foram criados. Os primeiros a serem apresentados são relacionados aos objetivos e tarefas dos agentes que compõem a base do *framework* JADE: *DFAgent* (responsável pelo serviço de "páginas amarelas"), *AMSAgent* (responsável pelo serviço de "páginas brancas") e o *Admin Agent* (responsável por controlar os demais agentes).

As Figuras 53 e 54 representam a interação do *AMSAgent* e do *Student Agent* que modela o serviço de registro dos agentes nas "páginas brancas" da plataforma. Exatamente como no modelo anterior, todo agente (representante de um usuário no sistema) registra-se na plataforma também no serviço de "páginas brancas".

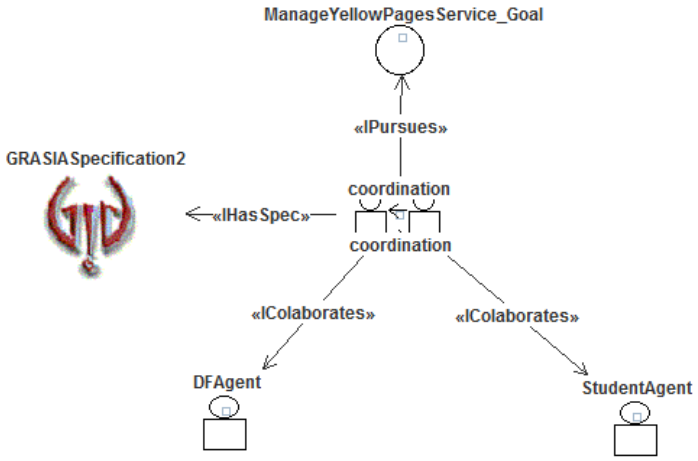


Figura 53: Modelo de Interação - Interação para o registro nas "páginas amarelas"

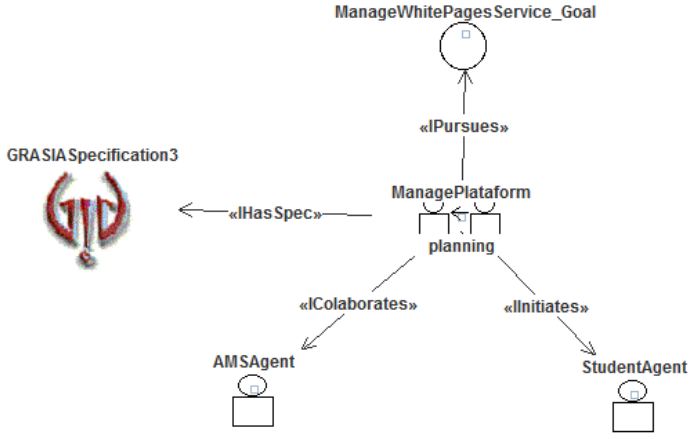


Figura 54: Modelo de Interação - Interação para o registro das "páginas brancas"

Na Figura 55 a interação entre o *DFAgent* e o *StudentAgent* é mostrada. Representa a entrada do usuário no sistema de busca dos objetos de aprendizagem e por conseqüência o agente que o representa sempre é registrado na plataforma pelo DFAgent.

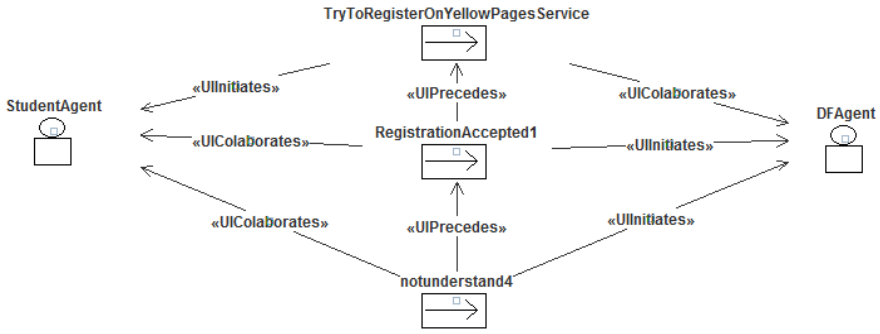


Figura 55: Modelo de Interação - Troca de mensagens entre o *DFAgent* e *StudentAgent*

Na Figura 56 a troca de mensagens entre os dois agentes é representada. A troca é composta de um pedido para registrar no serviço de "páginas amarelas" e a resposta com a confirmação do registro. No caso da mensagem conter algum erro, a resposta do *DFAgent* é uma mensagem "*NotUnderstand*" à requisição do *StudentAgent*.

A troca de mensagens também é semelhante: o *StudentAgent* tenta se registrar na plataforma no serviço provido pelo *AMSAgent*, em caso de sucesso ele recebe uma confirmação positiva e em caso da mensagem conter algum erro, o retorno do *AMSAgent* é uma mensagem de não entendimento do que foi requisitado.

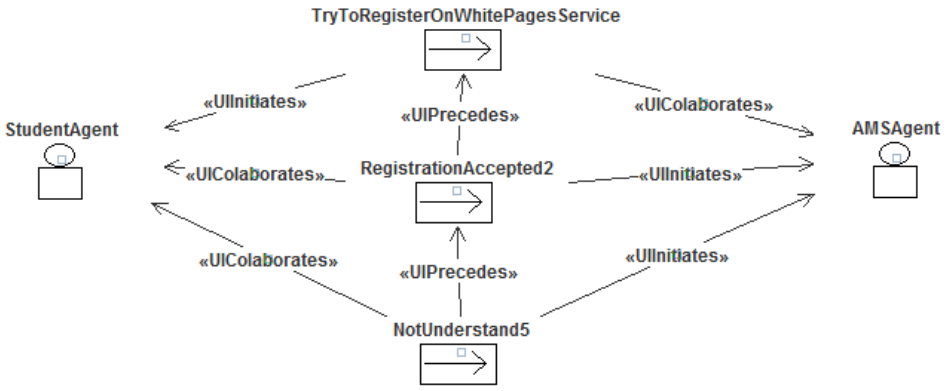


Figura 56: Modelo de Interação - Troca de mensagens entre o *AMSAgent* e o *StudentAgent*

Por último, no que diz respeito ao gerenciamento da plataforma, temos as interações do *Admin Agent*. As Figuras 57, 58 e 59 representam a interação dele com o *StudentAgent*, e depois, após receber a requisição de pesquisa, interage com o *PlannerAgent* para que o mesmo possa realizar a busca pelos objetos de aprendizagem, de acordo com os critérios passados pelo *Student Agent*. A Figura 60 representa a interação do *AdminAgent* com o *PlannerAgent.*

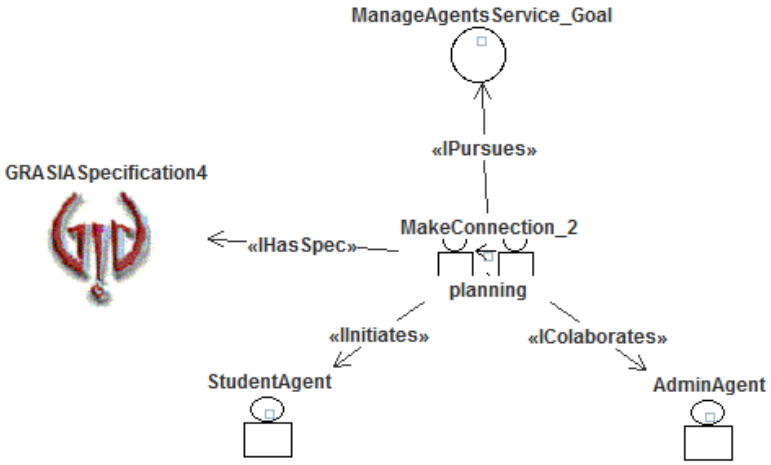


Figura 57: Modelo de Interação - Interação para a busca dos objetos de aprendizagem 1

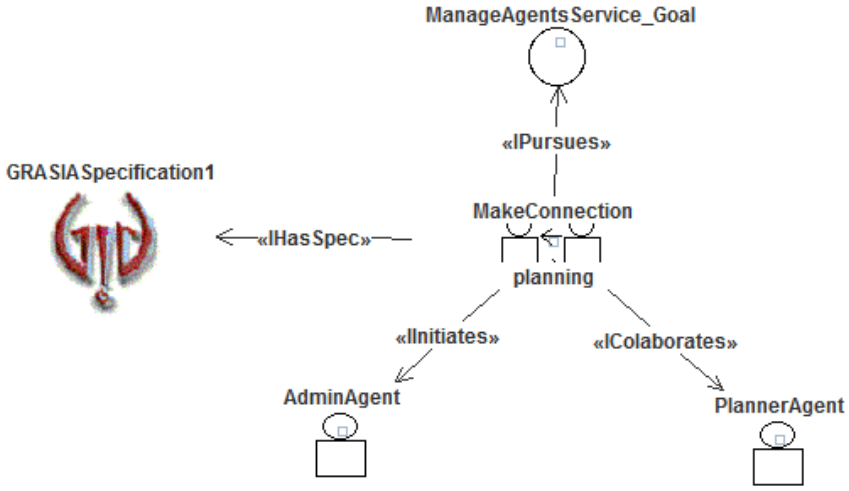


Figura 58: Modelo de Interação - Interação para a busca dos objetos de aprendizagem 2

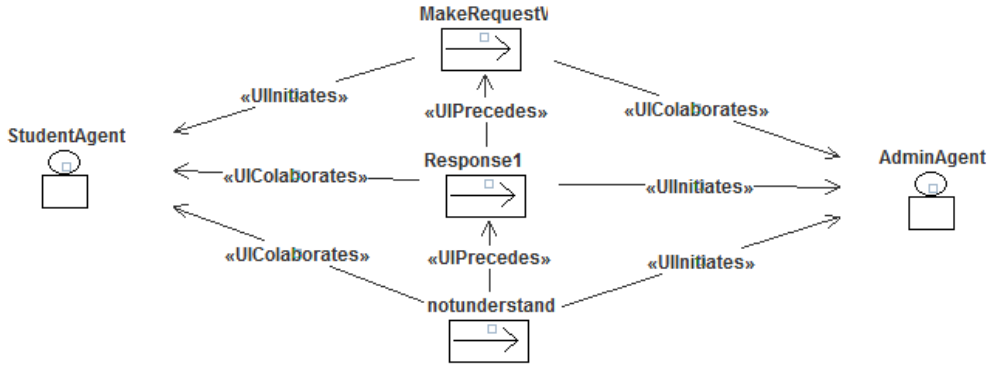


Figura 59: Modelo de Interação - Troca de mensagens entre *StudentAgent* e *AdminAgent*

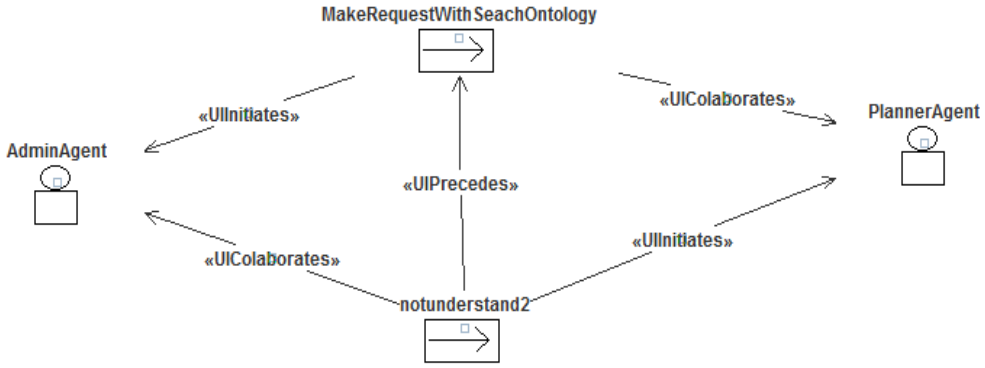


Figura 60: Modelo de Interação - Troca de mensagens entre *AdminAgent* e *PlannerAgent*

Os outros modelos de interação remetem ao objetivo de atender às necessidades dos usuários (representados pelo *StudentAgent*), quando for o caso. A Figura 61 e 62 representam a interação do *StudentAgent* (que possui o *PlannerAgent* como mediador). Os agentes buscam por objetos de aprendizagem utilizando a rede de ontologia.

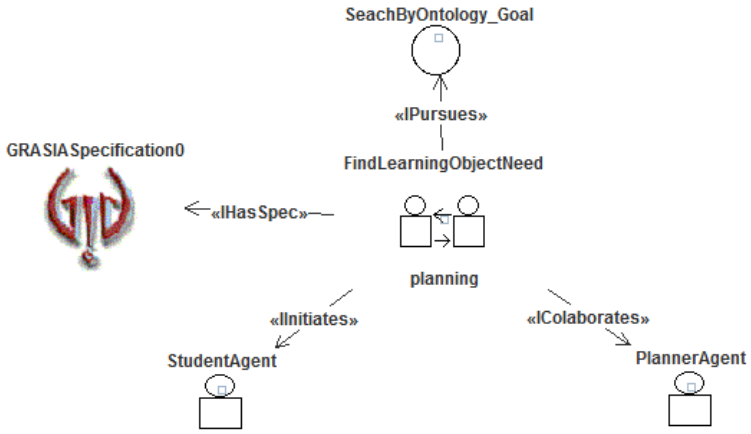


Figura 61: Modelo de Interação - Interação entre o *StudentAgent* e *PlannerAgent* para a busca dos OA

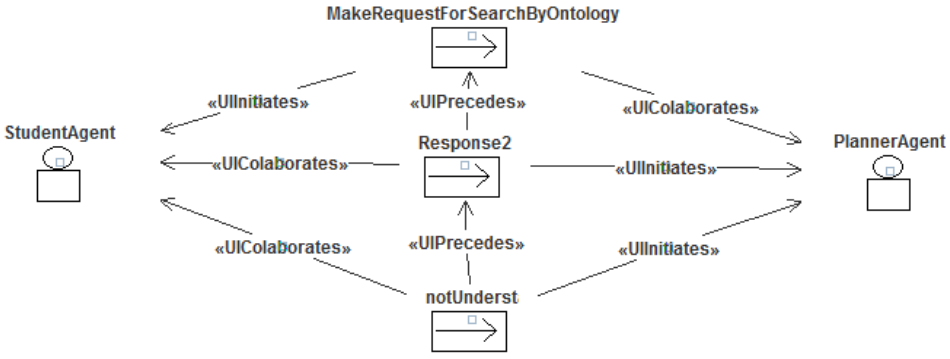


Figura 62: Modelo de Interação - Troca de mensagens entre o *StudentAgent* e *PlannerAgent*

6.4.1.6 MODELO DE AMBIENTE

Segundo Mendes (2009), o modelo de ambiente define a percepção do agente quanto a componentes existentes no sistema que não necessariamente sejam outros agentes. Uma base de dados, uma ontologia, uma aplicação externa, ou interna, até mesmo relação com um hardware, podem e devem ser modelados nesta seção.

Para esta plataforma foram identificados dois componentes presentes no ambiente dos agentes: o *Database*, responsável pelo armazenamento das informações em um banco de dados relacional, e a rede de ontologia. A Figura 63 ilustra o modelo de ambiente da plataforma.

Os três principais agentes (não nativos da plataforma JADE) se relacionam com todos os componentes. O *Planner Agent* utiliza o *Database* para armazenar as recomendações dos objetos de aprendizagem já realizadas, e quando outros agentes enviarem determinadas mensagens, ele usa as inferências das ontologias da rede de ontologias (*Ontologies Network*) para inferir novas recomendações.

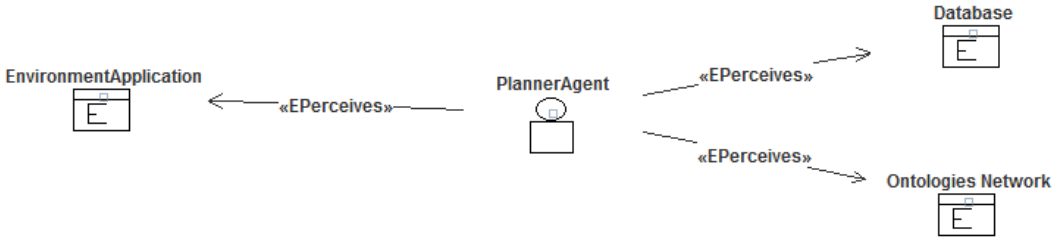


Figura 63: Modelo de Ambiente - Geral

6.5 ONTOLOGIA PERSONNA

As limitações observadas nos modelos de alunos apresentados neste trabalho motivaram o desenvolvimento de um modelo conceitual mais amplo, que fornecesse os principais dados dos Alunos, e que atendesse as necessidades dos sistemas de recomendação.

Uma das formas de obtenção das informações relativas ao perfil de aprendizagem do aluno normalmente é realizada através da monitoração das interações deste com o ambiente. Assim propomos a criação da ontologia PERSONNA baseada na ontologia de Dolog e Nejdl (2007) e Verbert et al. (2012). As ontologias apresentam uma estrutura mais promissora para utilização na modelagem de ambientes ubíquos e podem ser usadas para auxílio no critério de aplicabilidade a ambientes existentes.

A ontologia PERSONNA, que é uma das propostas deste trabalho tem por objetivo servir como base para modelagem do perfil e contexto do aluno em um ambiente *e-learning*, a fim de padronizar as informações e facilitar o processo de recomendação de objetos de aprendizagem da arquitetura BROAD-RS (Rezende et al. 2013). Ela foi especificada de forma

que possa ser aderente a qualquer ambiente de aprendizagem e, para este objetivo, consideramos os atributos mais relevantes dos alunos analizando os modelos existentes. Nesse contexto a ontologia PERSONNA avança nas pesquisas em relação à base conhecimento semântico usado em trabalhos prévios do grupo (Braga et al. 2011) (Campos et al. 2012b) (Santos, 2008b; Campos et al. 2011) (Souza et al., 2010).

A Figura 64 apresenta uma visão geral da ontologia PERSONNA com suas principais classes e propriedades.

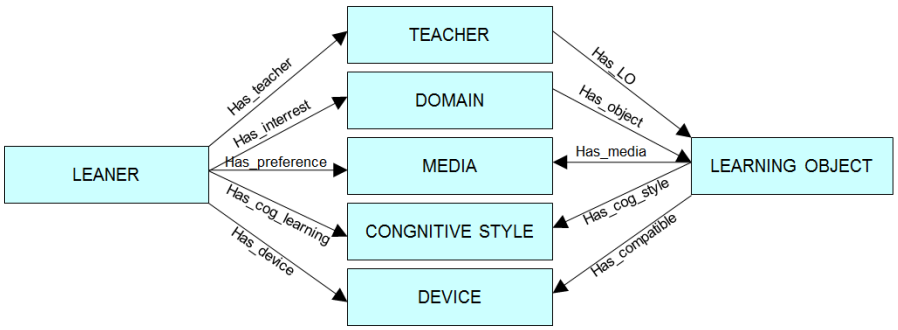


Figura 64: Visão geral da ontologia PERSONNA apresentando algumas de suas classes

A Tabela 13 apresenta as classes e detalhes da ontologia PERSONNA, listando suas classes, propriedades, tipo, a descrição do significado da propriedade e sua base de referência na literatura, onde se extraiu a classe.

Tabela 13: Categorias e Propriedades da Ontologia PERSONNA e seus detalhes

Classes	Propriedades	Tipo	Descrição	Referência
Learner Personal_Information (Identificação)	*Name*	*String*	Nome	(VERBERT et al. 2012) (DOLOG; NEJDL, 2007)
	Surname	*String*	Sobrenome	(VERBERT et al. 2012) (DOLOG; NEJDL, 2007)
	Sex	*Male / Female*	Sexo	(VERBERT et al. 2012) (DOLOG; NEJDL, 2007)
	Telephone	*Integer*	Telefone	(VERBERT et al. 2012) (DOLOG; NEJDL, 2007)
	Celphone	*Integer*	Telefone Celular	(VERBERT et al. 2012) (DOLOG; NEJDL, 2007)
	Email_contact	*String*	E-mail	(VERBERT et al. 2012) (DOLOG; NEJDL, 2007)
	Country	*Instance Data Country*	Pais/Nacionalidade	(VERBERT et al. 2012) (DOLOG; NEJDL, 2007)
	Postal_address	*String*	Endereço	(VERBERT et al. 2012) (DOLOG; NEJDL, 2007)
	Contacts	*String*	Informações de contato	(VERBERT et al. 2012) (DOLOG; NEJDL, 2007)
	Age	*Integer*	Idade	(VERBERT et al. 2012) (DOLOG; NEJDL, 2007)
	Social_security_num	*String*	CPF	(VERBERT et al. 2012)

	ber			(DOLOG; NEJDL, 2007)
	Password	*String*	Senha	(VERBERT et al. 2012)
	CognitiveStyle	*Visual / Textual / Auditory*	Estilo de aprendizagem: visual, textual, ou auditivo	(VERBERT et al. 2012)
	hasGoal	*Instance Goal*	Instância de Objetivos	Novo
	hasTimeNow	*Instance Time*	Instância de Tempo	Novo
	hasLocationNow	*Instance Location*	Instância de Localização	Novo
	hasPhysicalConditions	*Instance PhysicalConditions*	Instância de Condição Física	Novo
	hasSocialRelations	*Instance Social Relations*	Instância de Relacionamento	Novo
	hasPreferences	*Instance Preferences*	Instância de Preferência	Novo
	hasSpecialNeeds	*Instance Acessibility*	Instância de Condição Especial	Novo
	hasInterest	*InstanceInterest*	Instância de Interesse	Novo
	hasPerformance	*Instance Performance*	Instância de Desempenho	Novo
Performance (Desempenho)	*Performance_type*	*Nível do Curso*	Nível do Estudo	(DOLOG; NEJDL, 2007)
	Recording_date_time	*DateTime*	Data-hora do registro na base de dados	(DOLOG; NEJDL, 2007)
	Valid_date_time_begin	*DateTime*	Data e hora inicial de registros que são válidos por certo períodos de tempo, por exemplo: certificações	(DOLOG; NEJDL, 2007)
	Valid_date_time_end	*DateTime*	Data que expira o registro, por exemplo, data de expiração de uma certificação	(DOLOG; NEJDL, 2007)
	Issue_from_identifier	*String*	Responsável pela informação	(DOLOG; NEJDL, 2007)
	Issue_date_time	*DateTime*	Data e hora	(DOLOG; NEJDL, 2007)
	Issue_to_identifier	*String*	Identificador	(DOLOG; NEJDL, 2007)
	Learning_experience_identifier	*String*	Identificador da experiência do aluno	(DOLOG; NEJDL, 2007)
	Competency_identifier	*Integer*	Identificador associado com a definição de competência	(DOLOG; NEJDL, 2007)
	Performance_Granularity	*High / Mid / Low*	Granularidade	(DOLOG; NEJDL, 2007)
	Performance_coding_scheme	*String*	O tipo de codificação	(DOLOG; NEJDL, 2007)
	Performance_metric	*String*	Medidas	(DOLOG; NEJDL, 2007)
	Performance_value	*String*	Nota, valor de desempenho: o grau registrado Exemplo: "A", "4,3"	(DOLOG; NEJDL, 2007)
Acessibility (Acessibilidade)	*Acessibility_Comment*	*String*	Comentários sobre a categoria (idiomas de leitura, escrita e fala do aluno)	(VERBERT et al. 2012)
	Capabilities	*String*	Lista de capacidades	(VERBERT et al. 2012)

	Disabilities	*String*	Lista de incapacidades	(VERBERT et al. 2012)
	Acessibility_Content_type	*String*	O dado que é usado para descrever o recurso de acessibilidade	(VERBERT et al. 2012)
	AcessibilityLanguage	*Instance Data Language*	Idioma do recurso de acessibilidade	Novo
Preference (Informação de preferências)	*Language*	*Instance Data Language*	Idioma	(DOLOG; NEJDL, 2007)
	Country	*Instance Data Country*	País	(DOLOG; NEJDL, 2007)
	Media	*Tipo de Mídia*	Lista de preferências de mídias (texto, gráfico, áudio, vídeo, entre outros)	(DOLOG; NEJDL, 2007)
	CommunicarionTools	*Ferramentas de Comunicação*	Tipos de Comunicação Preferidas	(DOLOG; NEJDL, 2007)
	hasImportantOver	*Instance Preferences*	Ordem de Preferência	Novo
	hasDevices	*Instance Device*	Lista de preferências de dispositivos para interface homem-computador (*Tablets*, Celular, *SmartPhones* , PDA, PC, *Notebook*, TV Digital)	Novo
Interest (Interesses)	*Typename*	*String*	Tipo de interesse (hobbies, acadêmico, profissional)	(VERBERT et al. 2012)
	Comment	*String*	Comentários sobre a informação	(VERBERT et al. 2012)
	TAG	*String*	Dado usado para descrever o conteúdo	(VERBERT et al. 2012)
	Object	*String*	Objeto criado como parte de uma atividade de interesse	(VERBERT et al. 2012)
Social Relation (Relacionamento)	*SocialIdentifier*	*Integer*	Identificadores das pessoas que possuem algum tipo de relacionamento com o aluno	(VERBERT et al. 2012)
	RelationsLabel	*String*	Descreve a natureza do relacionamento (amigos, pessoas neutras, inimigos, vizinhos, colegas de trabalho e parentes)	(VERBERT et al. 2012)
	Relation_to_them	*String*	Como está relacionado com os outros	(VERBERT et al. 2012)
	Relation_to_me	*String*	Como os outros estão relacionados com ele	(VERBERT et al. 2012)
	Relation in SocialNetWork	*String*	Tipo de Rede Social (Nenhuma, Facebook®, Orkut®, Twitter®, entre outras)	Novo
Time (Tempo)	*Timenow*	*Time*	Tempo Atual	(VERBERT et al. 2012)
	Datenow	*Date*	Data Atual	(VERBERT et al. 2012)

	GMT	*Integer*	Fuso Horário	Novo
	FreeDay	*Boolean*	Feriado	Novo
	Time Comment	*String*	Comentários da informação	Novo
Tecnologic (Tecnológico)	*Devices*	*Tablets, Celular, PC, notebook, TV Digital*	Lista de preferências de dispositivos para interface homem-computador (Tablets, celular, PC, notebook, TV Digital)	(VERBERT et al. 2012)
	Devicetype	*Move / Fix*	Tipo de Dispositivo (Móvel ou Fixo)	(VERBERT et al. 2012)
	Bandwidth	*High / Low*	Largura de banda (Alta, Baixa)	(VERBERT et al. 2012)
	Browser	*String*	Navegador	(VERBERT et al. 2012)
	Browser Version	*Integer*	Versão do Navegador	(VERBERT et al. 2012)
	OS	*Instance OS*	Sistema Operacional	(VERBERT et al. 2012)
	Resolution	*String*	Resolução Ex: 800x600px	(VERBERT et al. 2012)
	OS Version	*Integer*	Versão do Sistema Operacional	(VERBERT et al. 2012)
	CPU	*String*	Desempenho da CPU em MHz	(VERBERT et al. 2012)
	GPU	*String*	Desempenho Gráfico	Novo
	CODEC List	*String*	Software *Codec* de Vídeos	Novo
	Comment	*String*	Comentários da informação	(VERBERT et al. 2012)
Goal (Objetivo)	*Typename*	*Professional / Educational / Personal*	Nome do objetivo	(VERBERT et al. 2012) (DOLOG; NEJDL, 2007)
	Comment	*String*	Comentários da informação	(VERBERT et al. 2012) (DOLOG; NEJDL, 2007)
	Contentype	*String*	Tipo de dado usado para descrever o conteúdo	(VERBERT et al. 2012) (DOLOG; NEJDL, 2007)
	Date	*Date*	Datas registradas para o objetivo	(VERBERT et al. 2012) (DOLOG; NEJDL, 2007)
	Priority	*High / Mid / Low*	Prioridade do objetivo	(VERBERT et al. 2012) (DOLOG; NEJDL, 2007)
	Status	*Boolean*	Status registrado do objetivo	(VERBERT et al. 2012) (DOLOG; NEJDL, 2007)
	Description	*String*	Descrição do objetivo	(VERBERT et al. 2012) (DOLOG; NEJDL, 2007)
	hasSubGoals	*Instance Goal*	Objetivos Secundários	(VERBERT et al. 2012) (DOLOG; NEJDL, 2007)
Location (Localização)	*IP*	*String*	IP do acesso	(VERBERT et al. 2012)
	Language	*Instance Data Language*	Idioma do local	(VERBERT et al. 2012)
	Location Country	*Instance Data Country*	Pais	(VERBERT et al. 2012)
	Location	*String*	Nome do Local	(VERBERT et al. 2012)
	GPS	*String*	Dados de sensor *Global Positiong System*	(VERBERT et al. 2012)
Physical Condition (Condição Física)	*Heat*	*High / Mid / Low*	Calor Ex: Alto, Médio, Baixo	(VERBERT et al. 2012)
	Light	*High / Mid / Low*	Luminosidade Ex: Alto, Médio, Baixo	(VERBERT et al. 2012)
	Sound	*High / Mid / Low*	Ruídos/Barulho Ex: Alto,	(VERBERT et al. 2012)

			Médio, Baixo	
	Weather	*String*	Tempo/Clima	(VERBERT et al. 2012)
	Comment	*String*	Comentários da informação	Novo

A Figura 65 apresenta a Ontologia PERSONNA aberta na Ferramenta Protegé e apresentando suas classes.

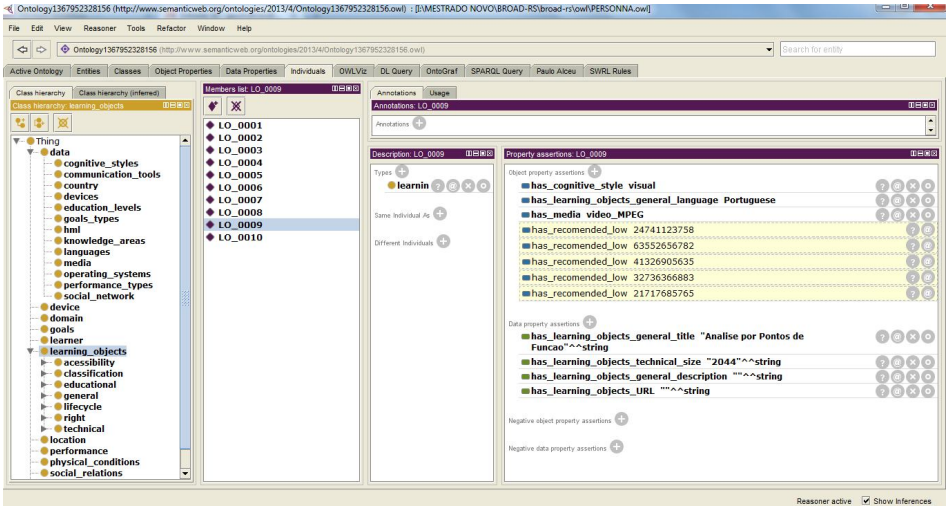


Figura 65: Ontologia PERSONNA aberta na Ferramenta Protegé

A Figura 66 apresenta a visão parcial das principais classes da ontologia PERSONNA usando a ferramenta VisGraph do Protégé.

Figura 66: Visão Parcial das Principais Classes da Ontologia PERSONNA

A Figura 67 apresenta a visão parcial das principais classes da ontologia de Learning Objects usando a ferramenta VisGraph do Protégé.

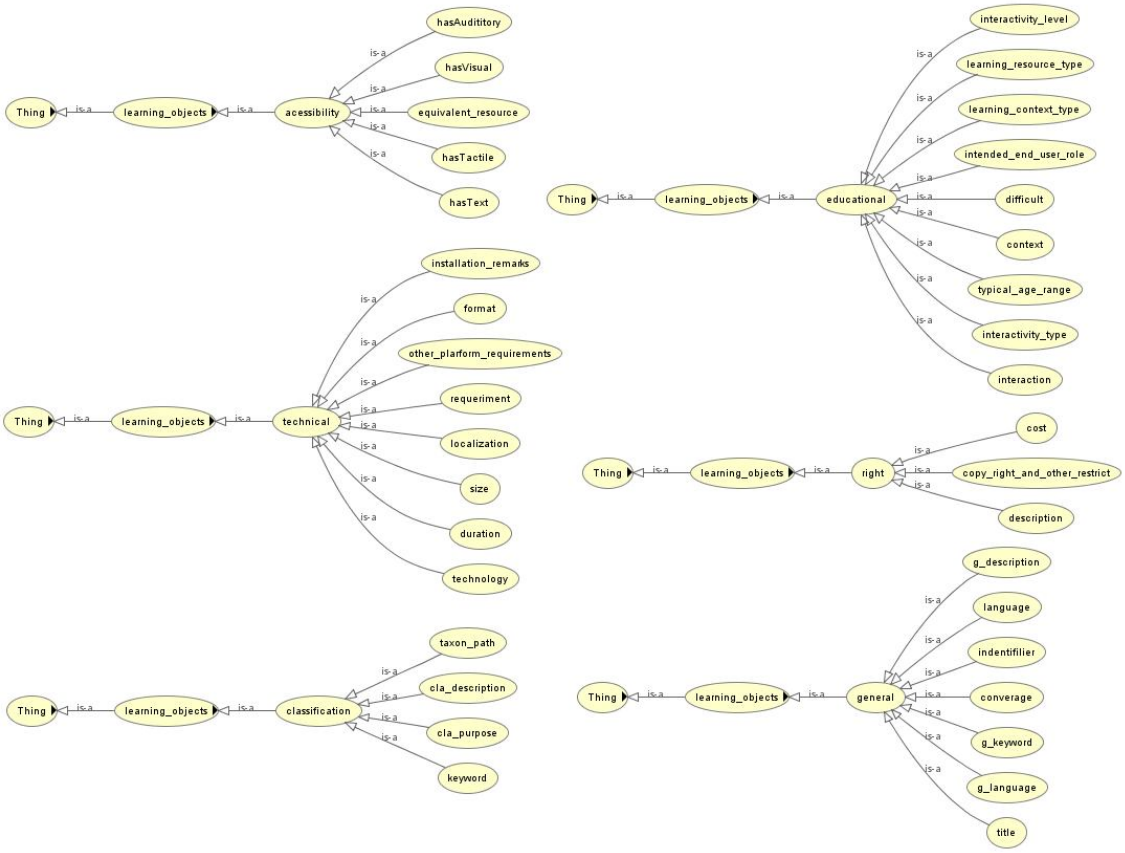


Figura 67 Visão Parcial das Principais Classes da Ontologia de Learning Objects

6.5.1 SITUAÇÕES DE APRENDIZAGEM

A situação de aprendizagem é definida como uma generalização que representa o contexto e perfil do aluno caracterizando suas preferências e conhecimento (quem?), a atividade educacional sendo realizada, a qual é internamente relacionada a uma determinada disciplina, projeto instrucional e objeto de aprendizagem sendo acessado (o que?), a tecnologia utilizada para acesso ao ambiente, que subentende a resolução do dispositivo, sistema operacional e taxa de conexão com a Internet (como?), ao local onde o aluno se encontra (onde?) e ao tempo corrente (quando?), entre outros.

Por exemplo, caso se deseje representar que: "um aluno está utilizando um dispositivo móvel", "ele está desenvolvendo atividades de estudo" e "ele está nesta mesma situação há quatro horas", é possível utilizar-se das expressões apresentadas na (Figura 68), as quais envolvem modelos ontológicos relativos: ao usuário, ao dispositivo computacional em uso; a uma atividade específica em curso, e ao tempo.

```
SituaçãoDeAprendizagem= {
    usa(OAluno.Aluno_X,ODispositivo.DispositivoMovel_Y),
    faz(OAluno.Aluno_X,OAtividades.Estudos),
    temSituacao(OAluno.Aluno_X,OSituacao.S_05),
    temInicio(OSituacao.S_05,OSituacao.13:00:00),
    temFim(OSituacao.S_05,OSituacao.17:00:00)
}
```

Figura 68. Generalização das situações de aprendizagem

6.5.2 REPRESENTAÇÃO DO CONTEXTO NA ONTOLOGIA

As relações existentes na rede de ontologias são representadas na forma de triplas ‹s, p, o›, onde o sujeito e o objeto representam instâncias da rede de ontologias, podendo pertencer às ontologias de mesmo domínio ou de domínios distintos.

Um predicado de contexto encapsula dois valores abstratos de contexto em uma relação, por exemplo: ‹{José}, acessa, {aula-01}›. Ele também pode ser representado por meio de variáveis, como será visto internamente às regras associadas às situações, por exemplo: ‹x, acessa, y›, onde x representa qualquer valor instanciado dentro do domínio Aluno e y qualquer valor instanciado dentro do domínio Educacional. Neste caso, qualquer par de valores em Aluno e OA.

Como um determinado par de valores que valida o predicado de contexto é válido em um tempo específico, um predicado de contexto instanciado é denotado por ‹‹s, p, o›, t›. Isso é importante para determinar o tempo em que determinados predicados de contexto que formam uma situação específica foram válidos. Para representar, por exemplo, que o aluno está na biblioteca neste momento: ‹{Aluno}, localizadoEm, Biblioteca ›, [2013-09-05 12:20:55] ›. Nesta representação, t pode ser um intervalo ou instante de tempo. Esta representação é possível de ser obtida por meio de OWL-DL, que possui construtores de classes relativamente ricos, permitindo alta expressividade para criação de classes e relacionamentos. Para raciocínio a respeito do conhecimento criado, a OWL permite determinação de restrições a respeito das classes e propriedades criadas, assim como verificação de inconsistências no conhecimento representado. Entretanto, o formalismo que define a linguagem OWL não permite a realização de raciocínio sobre as instâncias definidas na base de conhecimento, as quais constituem, neste trabalho, os predicados de contexto necessários para avaliação da situação do aluno.

Desta forma, optou-se por utilizar uma linguagem baseada em regras para determinação da situação do aluno, a qual é apresentada na próxima subseção.

6.5.3 FORMALIZAÇÃO DAS REGRAS DE INFERÊNCIA

Segundo Fleischmann (2012), a situação do aluno consiste ela própria de um predicado de contexto instanciado, pois: ‹{Aluno}, temSituacao, {S_01}›. Assim, tornou-se necessária a realização de raciocínio sobre os predicados de contexto válidos e, de acordo com este raciocínio, determinação da validade de um novo predicado de contexto, o qual representa a situação atual do aluno. Em caso de validade, este novo predicado passa a fazer parte da rede de ontologias.

O raciocínio a respeito das situações é realizado por meio da inter-relação entre o formalismo usado na definição da ontologia (Lógica Descritiva) e de regras definidas em SWRL (*Semantic Web Rule Language*). A SWRL consiste de uma linguagem de regras baseadas no formato das clausulas de Horn adicionado à OWL por meio de axiomas em formato OWL-DL, a qual permite a definição de raciocínio sobre as instâncias definidas na ontologia.

As regras em SWRL são definidas na forma de implicação entre um antecedente e um consequente. O seu significado determina que uma vez que as condições especificadas na parte antecedente forem atendidas, então as condições consequentes devem também ser atendidas. Antecedente e consequente são formados pela conjunção, na forma: $a1 \wedge a2 \wedge \ldots \wedge an \rightarrow C$. Os átomos presentes nas regras SWRL podem ser da forma: C(x), P(x, y), Q(x, y), *sameAs*(x, y) ou *differentFrom*(x, y), onde: C é uma classe em OWL-DL: P é uma propriedade de valores de indivíduos (objeto): Q é uma propriedade de valores de dados: x, y podem ser variáveis ou indivíduos OWL: e z pode ser uma variável ou valor de dado (fixo).

Para facilitar a compreensão da forma com que uma situação é representada em SWRL, Fleischmann (2012) apresenta uma regra de exemplo por meio de notação em Lógica de 1ª Ordem. Como a SWRL consiste em uma extensão da semântica apresentada na OWL-DL, onde o modelo teórico da linguagem OWL é interpretado por meio da seguinte abstração I:

I = ‹R, EC, ER, L, S, LV›

Onde:

- R consiste de um conjunto de recursos;
- LV consiste de um conjunto de valores literais ($LV \subseteq R$);
- EC é o mapeamento das classes e dos tipos de dados para subconjuntos em R e
- LV, respectivamente;
- ER é o mapeamento das propriedades para relações binárias em R;

- L é o mapeamento dos tipos literais para elementos de LV;
- S é o mapeamento dos nomes de indivíduos para elementos em EC(owl: Thing) (FLEISCHMANN, 2012).

A Figura 69 apresenta uma amostra das regras SWRL da ontologia PERSONNA na arquitetura BROAD-RS.

has_compatibled(?o, ?dv), has_device(?l, ?dv), has_interrest(?l, ?d), has_objects(?d, ?o) -> has_recomended_high(?o, ?l)
has_cognitive_style(?o, ?cs), has_interrest(?l, ?d), has_learner_cognitive_styles(?l, ?cs), has_objects(?d, ?o) -> has_recomended_high(?o, ?l)
learner(?p), integer[>= 18 , <= 65](?age), has_learner_age(?p, ?age) -> has_learner_driver_age(?p, true)
has_compatibled(?o, ?dv), has_device(?l, ?dv), has_objects(?d, ?o) -> has_recomended_mid(?o, ?l)
has_interrest(?l, ?d), has_objects(?d, ?o) -> has_recomended_low(?o, ?l)
has_compatibled(?o, ?dv), has_device(?l, ?dv) -> has_recomended_low(?o, ?l)
has_cognitive_style(?o, ?cs), has_learner_cognitive_styles(?l, ?cs) -> has_recomended_low(?o, ?l)
has_cognitive_style(?o, ?cs), has_learner_cognitive_styles(?l, ?cs), has_objects(?d, ?o) -> has_recomended_mid(?o, ?l)
has_learner_preferences_media(?l, ?me), has_media(?o, ?me), has_objects(?d, ?o) -> has_recomended_mid(?o, ?l)
has_interrest(?l, ?d), has_learner_preferences_media(?l, ?me), has_media(?o, ?me), has_objects(?d, ?o) -> has_recomended_high(?o, ?l)

Figura 69: Amostra das Regras SWRL da Ontologia PERSONNA

As regras SWRL definidas na ontologia PERSONNA são usadas para inferir o nível de aderência do aluno a um determinado objeto de aprendizagem de acordo com seu perfil, contexto e o plano didático definido pelo Professor. Para esse objetivo foram adotados três níveis:

- *High* ≥ 75% dos objetos selecionados atendem a um estilo de aprendizagem – totalmente aderente ao perfil e contexto do aluno.
- *Mid* ≥ 50% e < 75% dos objetos selecionados atendem a um estilo de aprendizagem - aderente ao perfil e contexto do aluno.
- *Low* < 50% dos objetos selecionados atendem a um estilo de aprendizagem - perfil e contexto não definido do aluno.

Para a ontologia PERSONNA foi necessário avaliar as porcentagens das relações entre

as propriedades (*has_compatibled* – Tem compatibilidade, *has_bandwidth_compatible* – Tem tamanho de banda compatível, *has_browser_compatible* – Tem navegador compatível, *has_interrest* – Tem Interesse, *has_goals* – Tem Objetivos, *has_preference* – Tem Preferências, entre outras) para que as regras de aderência fossem construídas (*High*, *Mid* e *Low*). Assim o agente *Planner Agent* da arquitetura, quando necessário, executa a máquina de inferência, utilizando a ontologia, apresentando como resultado as recomendações necessárias para cada aluno.

A Tabela 14 apresenta as relações de porcentagem para construção das regras para definição dos três níveis de aderência, consideramos neste exemplo algumas propriedades para exemplificação.

Tabela 14: Parte da Tabela de Relações de Porcentagem para contrução de regras

Nível	**has_teachers**	**has_cog_styles**	**has_devices**	**has_interrest**	**has_goals**	**has_preference**	**%**
HIGH	F	V	V	V	V	V	83,33%
	V	F	V	V	V	V	83,33%
	V	V	V	V	F	V	83,33%
	V	V	V	V	V	F	83,33%
MID	F	F	V	V	V	V	67,00%
	F	V	F	V	V	V	67,00%
	F	V	V	F	V	V	67,00%
	F	V	V	V	F	V	67,00%
	V	F	F	V	V	F	50,00%
	V	V	F	F	F	V	50,00%
	V	V	F	F	V	F	50,00%
	V	V	V	F	F	F	50,00%
LOW	F	F	F	F	V	V	33,34%
	F	F	F	V	F	V	33,34%
	F	V	F	F	F	F	16.35%
	V	F	F	F	F	F	16.35%

Para estabelecermos um sistema de recomendação através do sistema multi-agentes contamos com uma arquitetura estruturada, dispondo-se da rede de ontologias desenvolvida em linguagem OWL-DL, incluindo a ontologia PERSONNA, e suas regras previamente testadas e funcionais (em linguagem SWRL).

6.6 PROTÓTIPO

O Prototipo foi desenvolvido em linguagem JAVA (GOSLING, 2011) utilizando as

tecnologias JADE (BELLIFEMINE, 2003) para agentes e JENA para trabalhar com as ontologias (arquivos OWL). O *Reasoner* utilizado foi o *Pellet* e *OWLAPI* e comandos *SPARQL* para trabahar com os arquivos OWL das ontologias. A seguir são apresentadas algumas telas do protótipo desenvolvido.

A Figura 70 ilustra a tela de desenvolvimento do sistema da linguagem JAVA na plataforma *NetBeans.* Foram utilizadas também tecnologias como *Hibernate*, *Framework* ZK *Boss* e banco de dados *MySQL*.

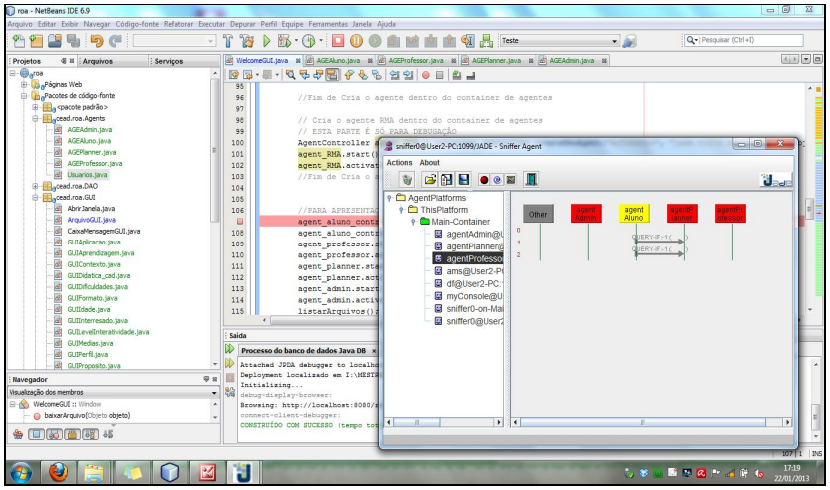


Figura 70: Tela do acompanhamento dos agentes no desenvolvimento do sistema

Algumas ferramentas que auxiliaram o desenvolvimento do protótipo foram o Protégé para trabalhar com ontologias OWL, o NetBeans como IDE da linguagem JAVA e Phpmyadmin para a estruturação do banco de dados relacional. A Figura 71 ilustra uma das ferramentas do JADE – O RMA (*Remote Manager Agents)*, uma plataforma própria para execução dos agentes e interface gráfica para gerenciamento.

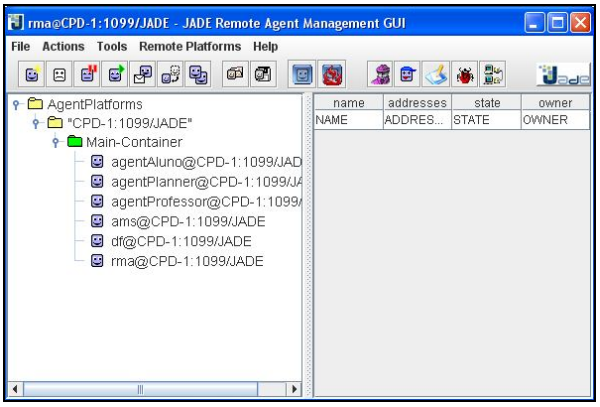


Figura 71: O RMA (*Remote Manager Agents*)

6.7 CENÁRIOS DE APRENDIZAGEM

Cenários são muito explorados pela Web Semântica na demonstração da operação prática de sistemas. Assim, esta seção descreve um cenário de aprendizagem típico, consistindo de diferentes situações internas sendo iniciadas e finalizadas através de eventos, resultado de ações do aluno sobre o ambiente educacional ou vindas do próprio ambiente. O cenário apresentado pré-entende a sensibilidade à situação de aprendizagem do aluno por parte do ambiente, mostrando quais elementos contextuais devem ser analisados a cada momento e as possíveis recomendações realizadas.

Esta seção descreve cenários de aprendizagem para a arquitetura BROAD-RS. Inicialmente, descreve-se o cenário de aprendizagem do início ao fim, para facilitar a compreensão. Nas próximas seções, este cenário é analisado em diferentes aspectos, sendo retomado passo a passo, explicando cada situação configurada, os elementos contextuais identificados e os aspectos dinâmicos observados em cada uma das situações (FLEISCHMANN, 2012).

O objetivo é apresentar de forma contextualizada as abordagens do projeto através de diagrama de sequência. Essa é baseada em um conjunto de passos realizados para cumprir determinada tarefa, além de mostrar quais atores e recursos estão envolvidos.

6.7.1 CENÁRIO 1 –PROFESSOR/PLANO DIDÁTICO E ALUNO/CONTEXTO

João é professor de Engenharia de Software do curso de Ciência da Computação e pretende usar o protótipo BROAD-RS em sua disciplina de Engenharia de Software para melhorar a qualidade do curso e fazer recomendação de objetos de aprendizagem aos alunos que atendam aos seus perfis.

Em uma Quinta-Feira o Professor João acessa o BROAD-RS usando usa identificação (*Username* e *Password*) (Figura 72).

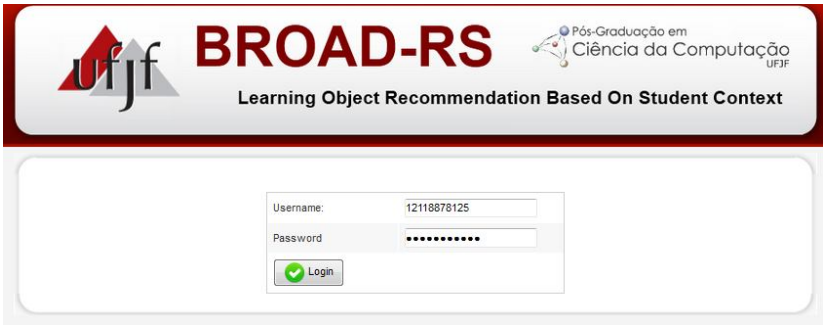


Figura 72: Tela de Login do BROAD-RS

Ao acessar o BROAD-RS é apresentada para o Professor João uma tela de “Bem vindo” identificando seu nome e tipo de usuário como Professor. Também é apresentado um “Menu” lateral direito com as opções do sistema: *Learning Objects* (Cadastro de Objetos de Aprendizagem), *My Didactic Plan* (Tela de Definição dos Planos Didáticos do Professor), *Update My Password* (Alteração de Senha) e *Exit* (Sair do Sistema) (Figura 73).

Figura 73: Tela de entrada do sistema com acesso tipo Professor

Como opção o Professor João pode alterar sua senha no protótipo como apresentado na Figura 74.

Figura 74: Tela de Alterar Senha

João sabe que o protótipo BROAD-RS é também um repositório de Objetos de Aprendizagem e pretende acrescentar novos OA para serem compartilhados para todos os usuários. Assim, para sua disciplina, o Professor João criou cinco OA para o sistema de recomendação BROAD-RS sendo: dois artigos científicos “ETL_Artigo” e “t9_2012_Ontological.pdf”. As Figuras 75, 76 e 77 apresentam o cadastro do objeto de aprendizagem “t9_2012_Ontological.pdf” com seus respectivos. Além disso o Professor João cadastrou uma vídeo-aula em alta resolução “0002002001.mp4”, um exercício “t9_2012_exercices.doc” e uma apresentação “Apresentation_ETL.doc”.

Também, na ferramenta BROAD-RS, o Professor João define seus planos didáticos para os objetos de aprendizagem de seu interesse. Os objetos não analisados no plano didático do professor João não são considerados para seus alunos. A Figura 78 apresenta o Professor João definindo o plano didático para o OA “t9_2012_Ontological.pdf”.

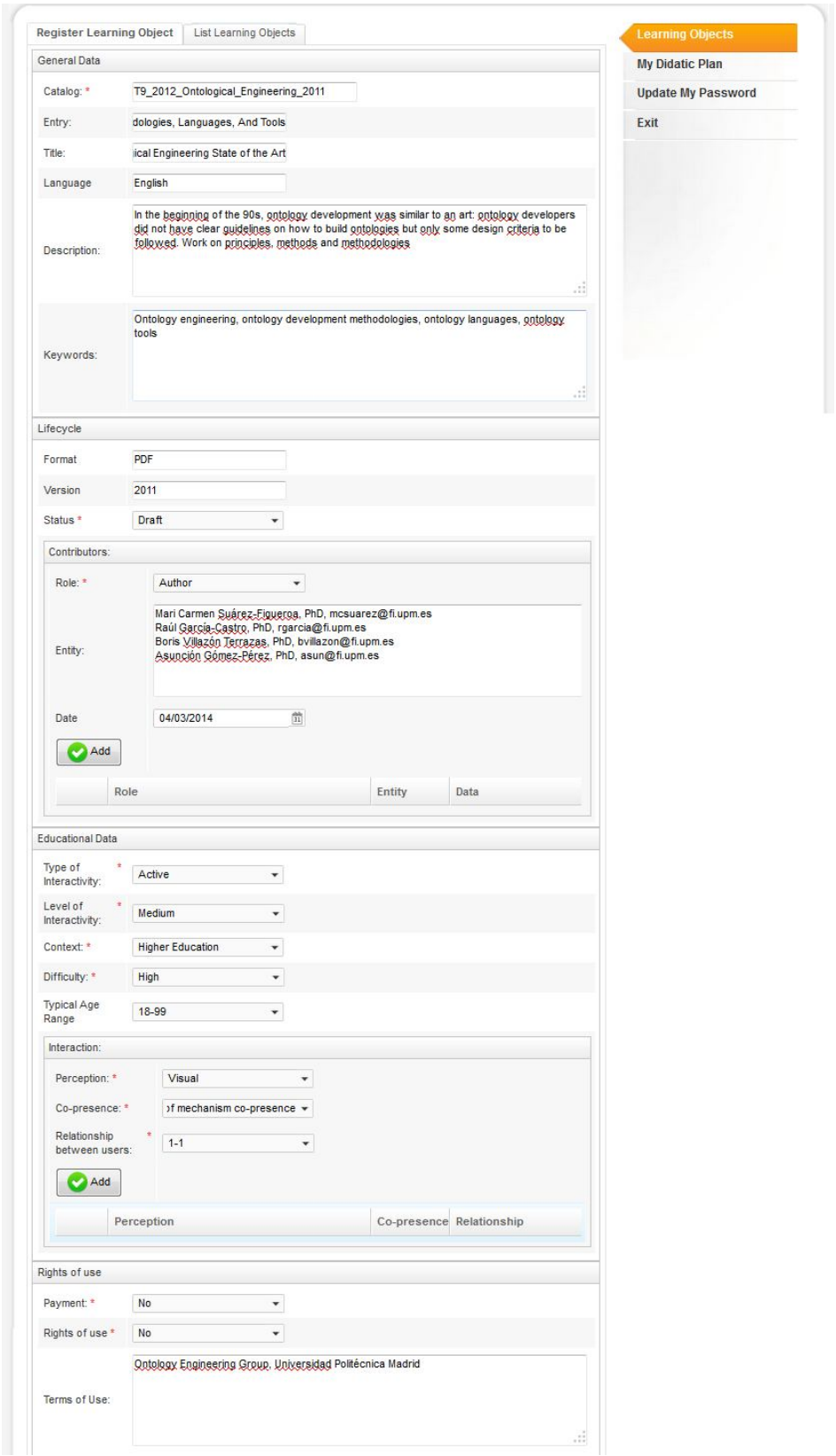


Figura 75: Cadastro de Objetos de Aprendizagem (Parte 1 de 3)

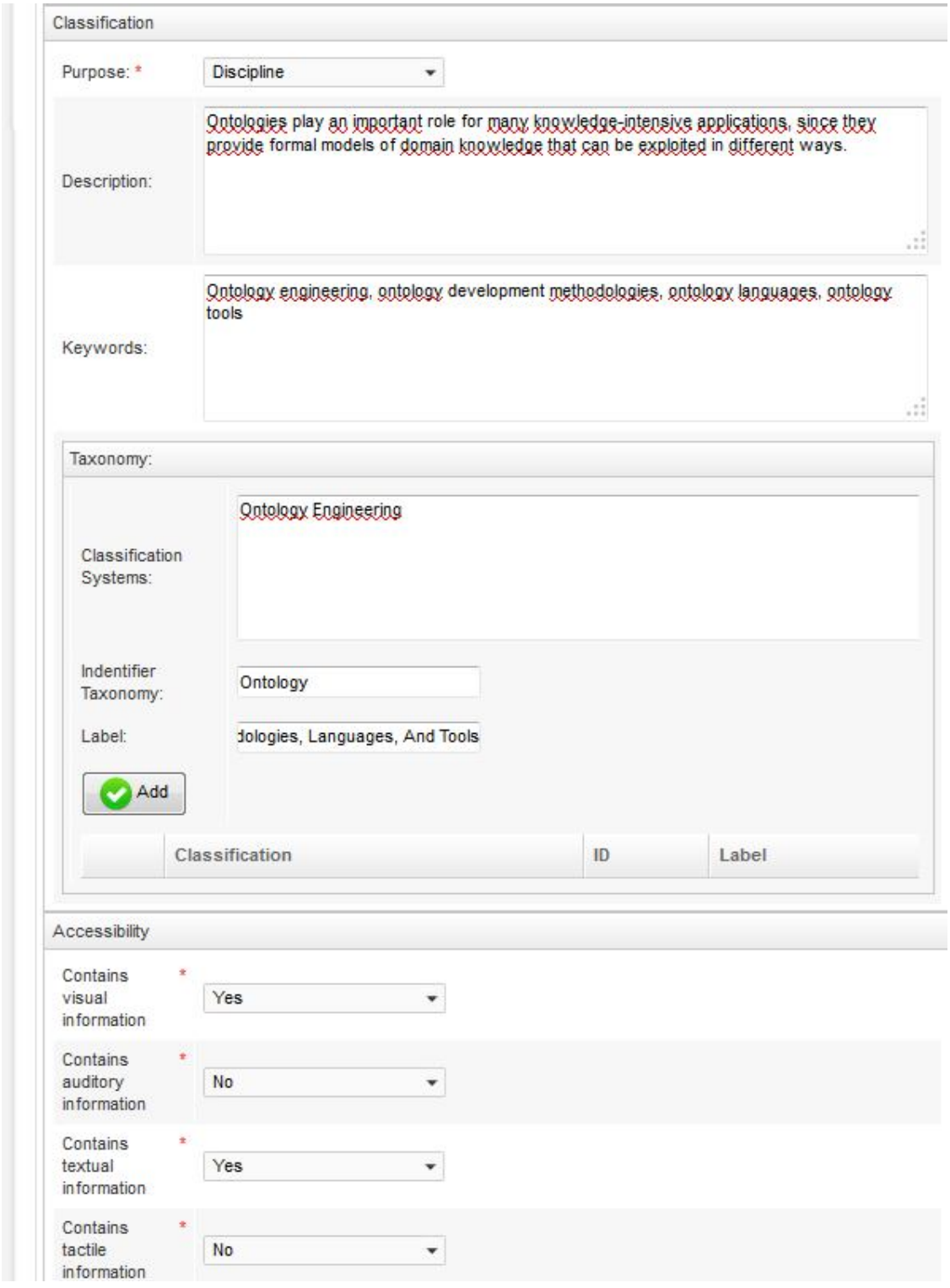


Figura 76: Cadastro de Objetos de Aprendizagem (Parte 2 de 3)

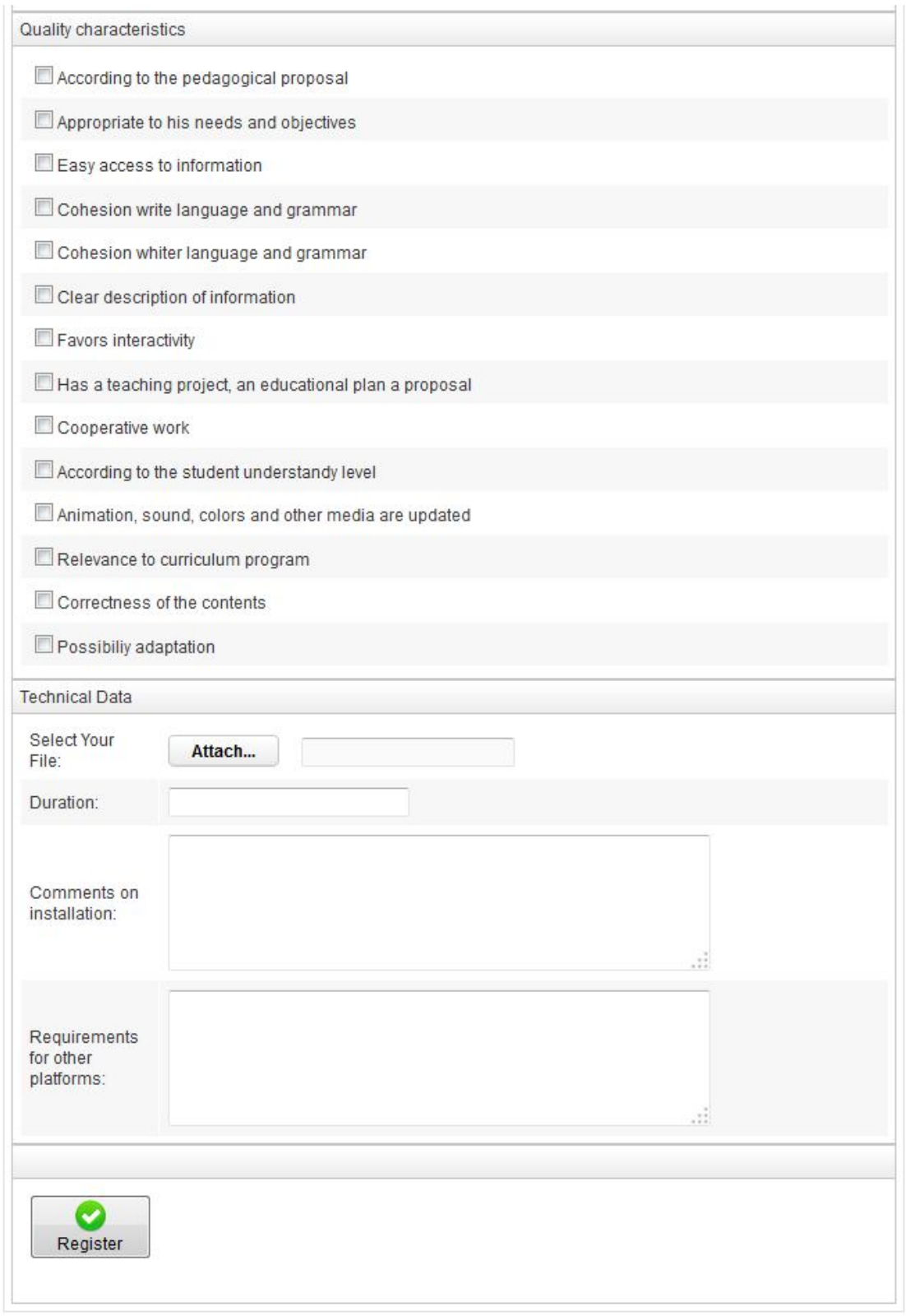


Figura 77: Cadastro de Objetos de Aprendizagem (Parte 3 de 3)

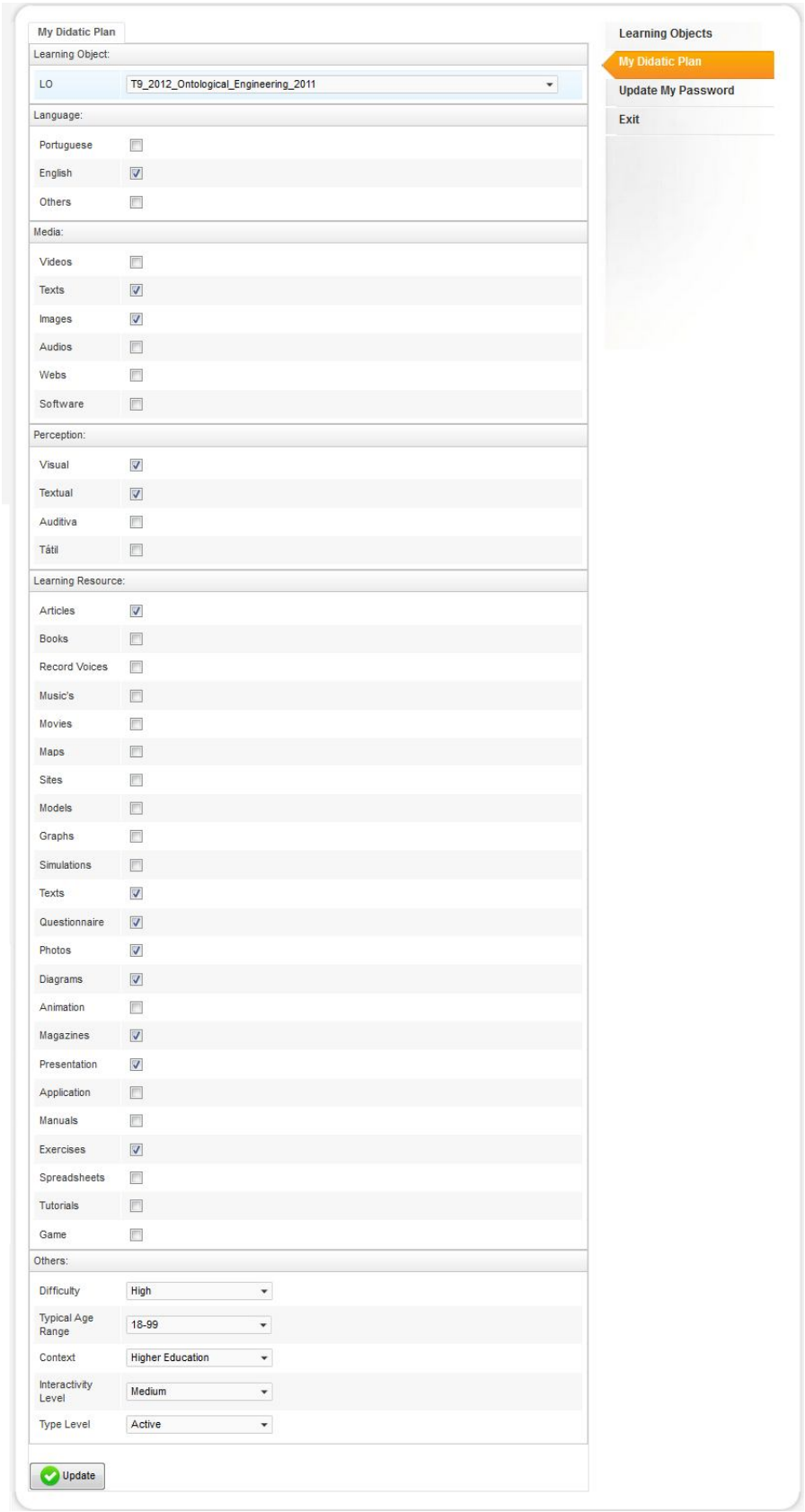


Figura 78: Tela de Definição do Plano Didático do Professor

Raquel é aluna do 4º ano do curso de Ciência da Computação e está cursando a disciplina de Engenharia de Software lecionada pelo Professor João.

Raquel domina a língua Portuguesa e Inglesa e apresenta comportamento ativo (aprende por meio de tentativas e experimentações) e visual (tem preferência por materiais em representação visual, como imagens, figuras, vídeos, diagramas e fluxos de dados).

É manhã de sexta-feira (11h) e Raquel está na biblioteca da universidade, acessando o sistema de recomendação como o Professor João havia solicitado. Ao entrar no protótipo é apresentada uma tela de "Bem vindo" identificando seu usuário e tipo como Aluna (Figura 79). Também é apresentado um "Menu" lateral direito com as opções do sistema: *My Recommendations* (Onde é Apresentada as Recomendações de OA), *My Profile/Context* (Definição dos dados de Contexto e Perfil do Aluno), *Update My Password* (Alteração de Senha) e *Exit* (Sair do Sistema).

Figura 79: Tela de Entrada do Sistema com Acesso tipo Aluno

A Figura 80 apresenta uma tela de onde o aluna cadastra seu dados de perfil e contexto no protótipo respondendo a um questionário sobre suas características.

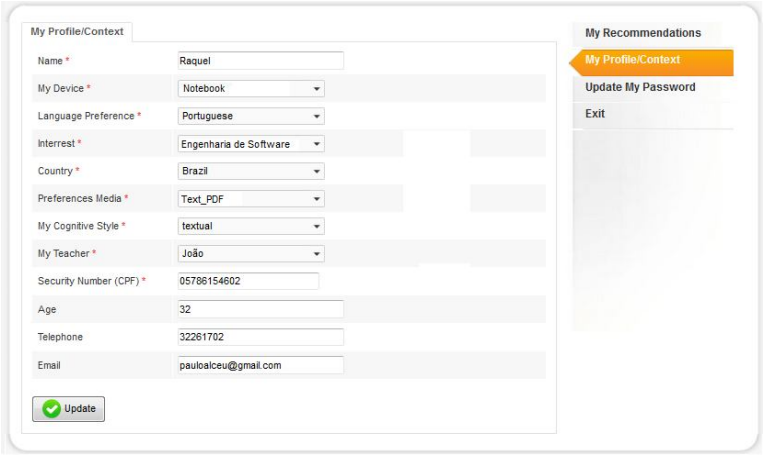


Figura 80: Tela de perfil e contexto do aluno (Perfil 1)

Raquel utiliza seu *Notebook*, com display de 15 polegadas (resolução 800x600), sistema operacional *Windows*, *browser* para navegação *Chrome*. Está conectado à rede da universidade por cabo (conexão fixa). Assim Raquel altera seu perfil para *My Device* = *Notebook*.

Raquel define que seu estilo de aprendizagem para textual. No momento sua preferencia de linguagem é Português e tipo de Arquivos PDF (*Portable Document Format*).

A Figura 81 apresenta as recomendações do protótipo para este momento em três níveis (*High, Mid e Low).* O sistema recomenda (*High*) um artigo e um livro considerando os dados de contexto e perfil definidos pela aluna. O sistema também recomenda (*Mid*) um vídeo e outro artigo por características de seus metadados e, recomenda (*Low*) outro artigo.

É importante considerar que o protótipo não recomenda OA que não foram avaliados no plano didático do Professor.

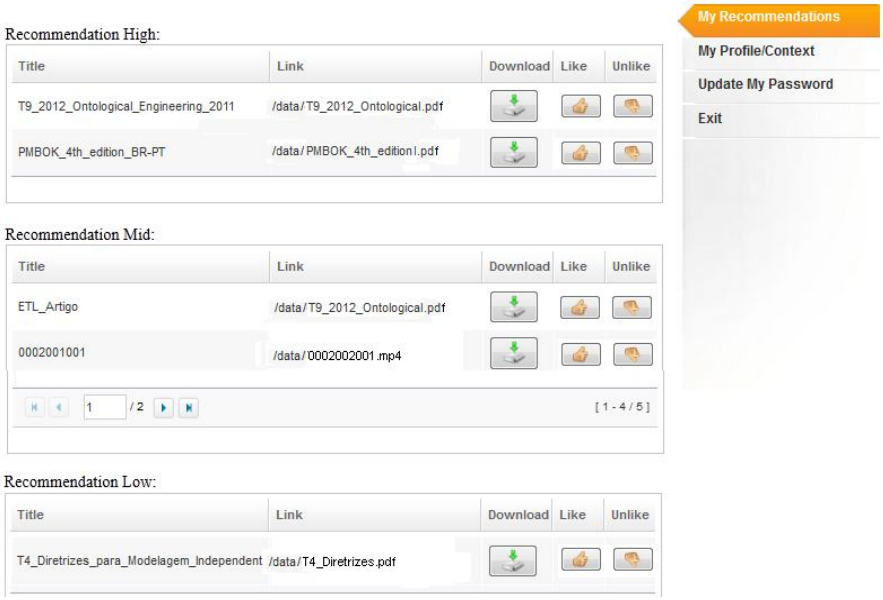


Figura 81: Tela de recomendação de OA do BROAD-RS (Primeira Recomendação)

Raquel sai da biblioteca (13h) e vai até a sala de aula, usa seu *smartphone* para acessar o sistema via rede wireless (rede móvel).

Raquel define que seu estilo de aprendizagem agora para visual. No momento sua preferencia de linguagem é Inglês e tipo de Arquivo MPEG *(Moving Picture Experts Group)* (Figura 82).

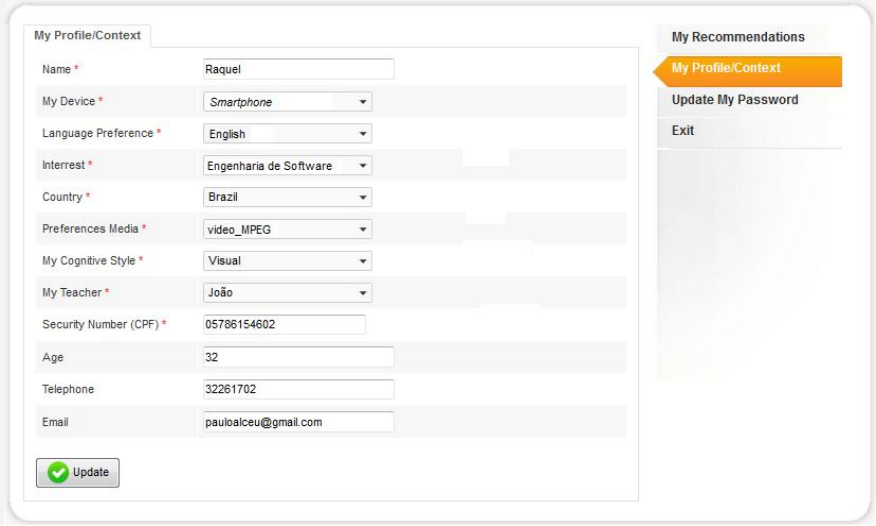


Figura 82: Tela de perfil e contexto do aluno (Perfil 2)

A Figura 83 apresenta a tela na qual estão sendo recomendados vários objetos de aprendizagem para a aluna Raquel. É neste momento que os agentes SA - *STUDENT AGENT* e PA - *PLANNER AGENT* entram em ação caso houver alguma alteração nos dados chamando a máquina de inferências das ontologias (*Peller Reasoner*). Então, neste segundo momento, o sistema recomenda (*High*) dois vídeos no formato MPEG considerando os dados de contexto e perfil definidos pela aluna. O Sistema também recomenda (*Mid*) um vídeo em MP4 por características de seus metadados e, recomenda (*Low*) outro artigo.

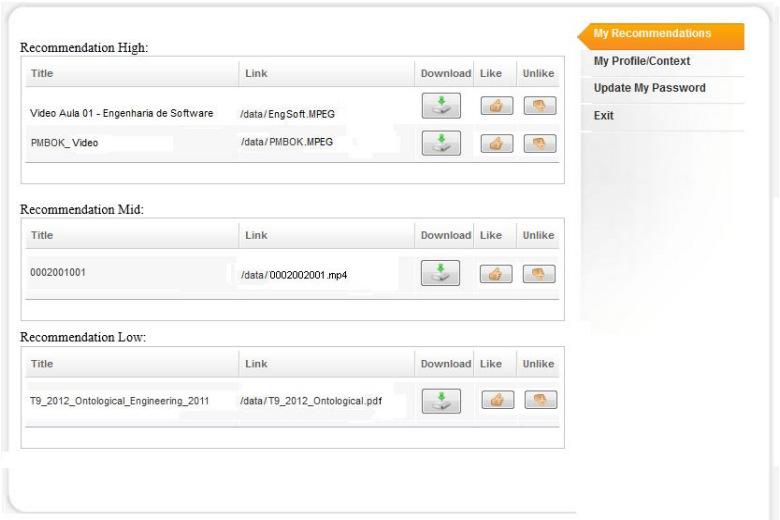


Figura 83: Tela de recomendação de OA do protótipo BROAD-RS (Segunda Recomendação)

6.8 ANÁLISE DO CENÁRIO EDUCACIONAL APRESENTADO

O cenário apresentado consiste em recomendações do OA, de uma disciplina de Engenharia de Software para alunos dos cursos de Computação. Neste cenário, vários OA são apresentados à aluna.

Para analisar o cenário apresentado com relação às situações configuradas, primeiramente são avaliados os elementos contextuais relevantes como importantes na definição da situação corrente:

1) Quem?

João, sexo masculino, professor do curso de Ciência da Computação, ministrando a disciplina de Engenharia de Software.

Raquel, sexo feminino, aluna do curso de Ciência da Computação, cursando a disciplina de Engenharia de Software que domina a língua inglesa (dados pessoais).

2) Como?

Aluna Raquel está usando *notebook* com tela resolução 800x600, sistema operacional Windows, browser de navegação com a *Internet Chrome* e conexão fixa a cabo e depois usando um *smartphone* via *wireless* (dados tecnológicos).

3) Quando?

O Professor João acessa o sistema BROAD-RS na Quinta-Feira. Já a aluna Raquel acessa o sistema às 11h e 13h de sexta-feira (dados temporais).

4) Onde?

Aluna Raquel na biblioteca da Universidade e, posteriormente, na sala de aula (dados de localização).

5) O que?

Desenvolvendo varias atividades da disciplina de Engenharia de Software (dados do domínio educacional).

De acordo com esta análise inicial, percebe-se que para definição da situação de aprendizagem na qual o aluno se apresenta deve-se analisar os seguintes grupos de elementos contextuais:

- Dados Pessoais e Comportamentais - nome, sexo, cursos matriculados, disciplina, línguas conhecidas (fluência), conhecimento, preferências no uso do ambiente educacional, e forma de navegação no conteúdo didático.
- Dados do Domínio - dados relativos aos OAs disponíveis ao aluno (domínio educacional) e modelo educacional criado pelo professor.
- Dados Tecnológicos - dados a respeito do dispositivo computacional utilizado para navegação no conteúdo didático.

- Dados de Localização - local onde o aluno está situado.
- Dados de Tempo - tempo corrente de acesso ao sistema educacional.

6.9 ANÁLISES RELATIVAS AOS AGENTES

6.9.1 CENÁRIO 2 – PROFESSOR ACESSANDO SISTEMA

No cenário descrito na Figura 84 é apresentado o processo que ocorre quando o professor acessa o sistema sem nenhuma modificação de dados.

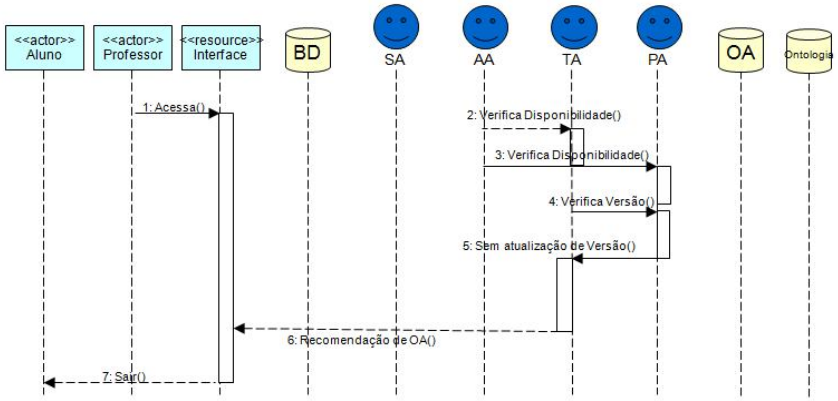


Figura 84: Cenário de Aprendizagem 2 - Professor acessando sistema

Inicialmente o professor acessa o sistema (1: Acessar()). O agente AA (*ADMINISTRATION AGENT*) verifica se o agente TA (*TEACHER AGENT*) do professor e o agente PA (*PLANNER AGENT*) estão em modo ativo (2: VerificarDisponibilidade() e 3: VerificarDisponibilidade()). O agente PA verifica se existe atualização nos dados (Plano Didático ou Objetos de Aprendizagem) (4: VerificarVersão(), 5: Sem atualização de Versão(). Como não houve atualização o agente PA notifica o Professor pela interface usando o banco de dados relacional (6: Recomendação de OA()).

6.9.2 CENÁRIO 3 – PROFESSOR FAZENDO ALTERAÇÕES NO SISTEMA (PLANO DIDÁTICO)

No cenário da Figura 85 é apresentado o processo que ocorre quando o professor acessa o sistema e relaciona seu plano didático aos objetos de aprendizagem.

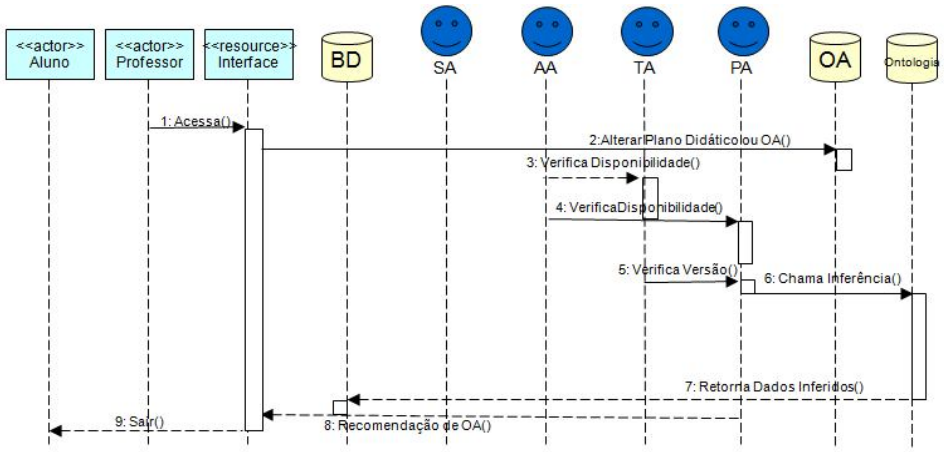


Figura 85: Cenário de Aprendizagem 3 - Professor criando seu plano didático

O usuário professor acessa o sistema (1: Acessar()). O professor tem a possibilidade de cadastrar objetos de aprendizagem e plano didático (2: Alterar Plano Didático ou OA()). O agente AA (*ADMINISTRATION AGENT*) verifica se o agente TA (*TEACHER AGENT*) do professor e o agente PA (*PLANNER AGENT*) estão em modo ativo (3: VerificarDisponibilidade() e 4: VerificarDisponibilidade()). O agente TA verifica se existe atualização nos dados (5: Verificar Versão()). Como houve atualização nos dados o agente PA chama a maquina de inferência da ontologia (6: Chama Inferência()). Assim os dados inferifos são carregados em um banco de dados relacional e a interface (7: Retorna Dados Inferidos() e (8: Recomendação de OA())).

6.9.3 CENÁRIO 4 – ALUNO FAZENDO ALTERAÇÕES NO SISTEMA (PERFIL E CONTEXTO)

No cenário da Figura 86 é apresentado o processo que ocorre quando o aluno acessa e cria ou altera seus dados de contexto e perfil.

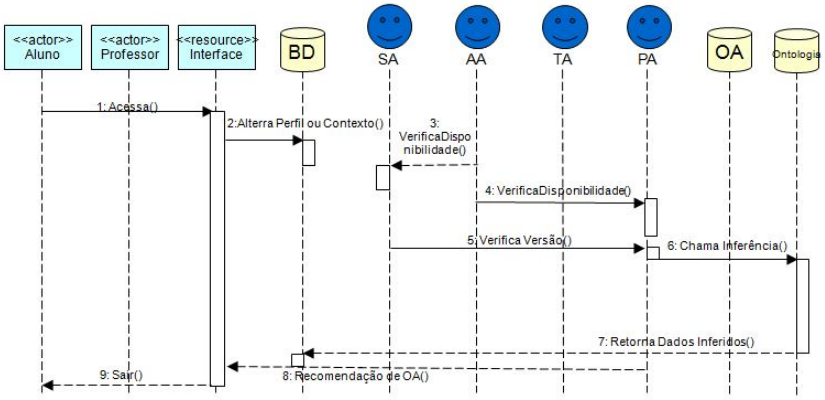


Figura 86: Cenário de Aprendizagem 4 - Aluno fazendo alterações no sistema (Perfil e

Contexto)

O usuário aluno acessa o sistema (1: Acessar()). O aluno tem a possibilidade de alterar ou criar seus dados de contexto e perfil (2: Alterar Perfil ou Contexto()). O agente AA (*ADMINISTRATION AGENT*) verifica se o agente SA (*STUDENT AGENT*) do professor e o agente PA (*PLANNER AGENT*) estão em modo ativo (3: VerificarDisponibilidade() e 4: VerificarDisponibilidade()). O agente SA verifica se existe atualização nos dados (5: Verificar Versão()). Como houve atualização nos dados o agente PA chama a máquina de inferência da ontologia (6: Chama Inferência()). Assim os dados inferifos são carregados em um banco de dados relacional e apresentados na interface (7: Retorna Dados Inferidos() e (8: Recomendação de OA())).

6.9.4 CENÁRIO 5 – ALUNO ACESSANDO O SISTEMA

No cenário da Figura 87 é apresentado o processo que ocorre quando o aluno acessa o sistema sem alterar seus dados.

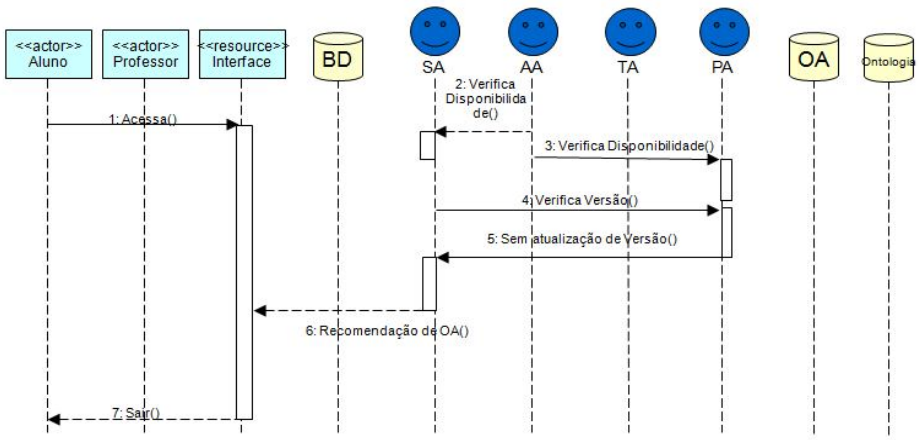


Figura 87: Cenário de Aprendizagem 5 - Aluno acessando o sistema

Inicialmente o aluno acessa o sistema (1: Acessar()). O agente AA verifica se o agente TA (*TEACHER AGENT*) do professor e o agente PA (*PLANNER AGENT*) estão em modo ativo (2: VerificarDisponibilidade() e 3: VerificarDisponibilidade()). O agente SA (*STUDENT AGENT*) verifica se existe atualização nos dados (Contexto e Perfil do aluno) (4: VerificarVersão(), 5: Sem atualização de Versão(). Como não houve atualização o agente PA notifica o Aluno pela interface usando o banco de dados relacional (6: Recomendação de OA()).

6.10CONSIDERAÇÕES FINAIS DO CAPÍTULO

Tendo em vista a enorme quantidade de OA e materiais que o aluno tem a sua disposição hoje

em dia, de diversos enfoques sobre um mesmo conceito, o uso de sistemas de recomendação se faz necessário, auxiliando o processo de busca e recuperação de OA e apresentando ao aluno os OA mais adequados ao seu perfil.

Este capítulo apresentou o histórico do projeto BROAD e descreveu a arquitetura BROAD-RS, sua estrutura, seus agentes e suas especificações, o plano didático, a estrutura da ontologia PERSONNA, o protótipo desenvolvido e alguns cenários de aprendizagem.

A arquitetura BROAD-RS de recomendação de objetos de aprendizagem permite a automação do processo conforme perfil e contexto do aluno e de acordo com o plano didático especificado pelo professor.

Neste capítulo tambem propusemos a criação da ontologia PERSONNA, que tem por objetivo servir como base para modelagem do perfil e contexto do aluno em um ambiente *e-learning*, a fim de padronizar as informações e facilitar o processo de recomendação de objetos de aprendizagem da arquitetura BROAD-RS. Ela foi especificada de forma que possa ser aderente a qualquer ambiente de aprendizagem e, para este objetivo, consideramos os atributos mais relevantes dos alunos analizando os modelos existentes.

Com a criação dos cinco cenários de aprendizagem obteve-se uma avaliação mais objetiva da proposta. É observada a relação automática dos agentes com a rede de ontologias. Quando os alunos e professores não alteram os dados, como o de contexto e plano didático, os agentes não necessitam inferir novas recomendações usando as ontologias pois as recomendações antigas estão armazenadas no banco de dados, porém em caso de alteração os agentes reagem às modificações de forma automática, e nova recomendações podem ser geradas. Esta estratégia é fundamental para que o processo de recomendação do BROAD-RS seja realizado de forma dinâmica e ágil.

7 CONSIDERAÇÕES FINAIS

A sensibilidade ao contexto vêm ganhando espaço no desenvolvimento de sistemas e aplicações que passam a apresentar ênfase na análise do contexto dos usuários e das situações configuradas no decorrer do tempo, buscando fazer com que os sistemas possam proativamente perceber as melhores condições de ação e agir de acordo com esta percepção.

Este trabalho abordou questões ligadas aos temas como recomendação, sensibilidade ao contexto no âmbito de sistemas *e-learning*, mais especificamente, sistemas de recomendação de objetos de aprendizagem sensíveis ao contexto. Assim, o desenvolvimento da pesquisa consistiu da ampla análise do contexto, com o intuito de investigar a hipótese proposta por este trabalho: a possibilidade de modelar e implementar um sistema de recomendação de OA de forma a aperfeiçoar as características de adaptabilidade aos perfis dos alunos.

Essa pesquisa contribuirá para ampliar a utilização de objetos de aprendizagem através de uma sistematização do processo de recomendação de conteúdo educacional aderente a um plano do professor e características dos alunos.

A arquitetura BROAD-RS foi modelada utilizando a metodologia de sistemas multi-agentes INGENIAS utilizando a ferramenta IDK (*INGENIAS Development Kit).* Com o desenvolvimento da arquitetura verificam-se as seguintes vantagens: é sensível ao contexto e perfil do aluno, é considerada a análise do professor (plano didático), utiliza de recursos semânticos como a utilização de uma rede de ontologia, todas as recomendações realizadas são armazenadas em banco de dados.

Para tanto, o intuito desta abordagem é aperfeiçoar o processo de ensino-aprendizagem de forma transparente aos estudantes, provocando assim um maior interesse no estudo.

Nesta arquitetura, os agentes podem requisitar serviços da camada de base de conhecimento, rede de ontologias, com capacidade para estabelecer dinamicamente a recomendação mais apropriada de acordo com as necessidades específicas do usuário.

Como trabalhos futuros há algumas melhorias que podem ser realizadas:

i) testar o ambiente desenvolvido em áreas específicas (ex. matemática, engenharia de software, biologia, administração, entre outras) para verificar a adequabilidade da recomendação de OAs em conteúdos específicos;

ii) considerar mais características de contexto do estudante antes de decidir se um

OA é adequado, aperfeiçoando, consequentemente, o mecanismo de recomendação.

iii) capturar informações de perfil e contexto do usuário em redes sociais e ambientes de aprendizagem.

iv) preparar esta arquitetura para uso em dispositivos móveis, como celulares, incluindo um novo agente de interface para a análise do ambiente a qual o OA é recomendado.

v) criar um agente que captura dados de contexto do ambiente como: localização GPS, tipo de equipamento, tipo de sistema operacional, entre outros.

vi) preparar esta arquitetura para se comunicar com outras arquiteturas que usam agentes, incluindo um novo agente de comunicação externa integrando cada vez mais a rede de ontologias.

A hipótese deste trabalho foi comprovada pela modelagem e implementação da arquitetura BROAD-RS e seus objetivos de recomendação de OA com características de adaptabilidade ao contexto e perfil do aluno. O modelo formal de uma situação de aprendizagem foi apresentado, assim como seus cenários de aprendizagem.

Outra hipótese, a identificação dos dados relevantes para definição da recomendação de objetos de aprendizagem, foi alcançada pela proposta da ontologia de contextos e perfil de alunos denominada PERSONNA.

A experiência obtida com o desenvolvimento deste trabalho, nos leva a concluir que o desenvolvimento de ambientes sensíveis ao contexto tende a exigir mais do professor no que diz respeito ao desenvolvimento de OAs mais ricos, pois precisam prever diferentes situações de aprendizagem e buscar meios mais adequados para apresentação do conteúdo educacional. Por isso, ferramentas de auxílio à autoria se fazem cada vez mais necessárias, de forma a facilitar o desenvolvimento desses objetos de aprendizagem.

Finalmente, entende-se que apesar de existirem limitações nos dias atuais relativas à falta de flexibilidade dos ambientes de aprendizagem e falta de ferramentas para auxílio à autoria de conteúdo educacional, o futuro caminha para o desenvolvimento destes ambientes, mais ricos, adaptativos e flexíveis às diversas situações enfrentadas pelos alunos em suas atividades didáticas diárias.

REFERÊNCIAS

ADOMAVICIUS, G.; TUZHILIN, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Knowledge and Data Engineering, In: IEEE Transactions on, EUA, 2005.

ALHARBI, M. T, et al. Context-aware Personal Learning Environment. The 7th International Conference for Internet Technology and Secured Transactions, Leicester, UK, 2012.

ANAND, S. R.; GEORGEFF, M. P. BDI Agents: from Theory to Practice, In: Proceedings of the First International Conference on Multi Agent Systems, San Francisco, USA, 1995.

ARIADNE. Ariadne - Educational Metadata Recommendation, Disponível em: <http://vs.fernuni-hagen.de/methoden/ils/Organisation/ariadne.html>. Acesso em: 5 de setembro de 2012. Leuven, Belgium, 2002.

BAZIRE, M.; BRÉZILLON, P. Understanding Context Before Using It, In: 5th International and Interdisciplinary Conference, CONTEXT-05, v. LNAI 3554, Springer Verlag, Paris, France, 2005. p. 29-40.

BEHAR, P. A. Modelos Pedagógicos em Educação a Distância. In: Instituto de Informática – Universidade Federal do Rio Grande do Sul - UFRGS, Porto Alegre, RS, Brasil, 2009.

BEHR, A.; PRIMO, T.; VICCARI, R. An Ontology for the OBAA Metadata Standard. In: Simpósio Brasileiro de Informática na Educação - SBIE, Anais do XXIII Simpósio Brasileiro de Informática na Educação, Rio de Janeiro, Brasil, 2012.

BELLIFEMINE, F. JADE a White Paper, In: Rel. Téc., 2003. exp - Volume 3 - n. 3 - Telecom, Italia.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The semantic web, In: Scientific American, Publicação Eletrônica, San Francisco, EUA, 2001.

BORGES, F.; NAVARRO, M. Aplicação colaborativa de objetos de aprendizagem, a partir de uma proposta de planejamento pedagógico integrado, Belo Horizonte, Brasil, 2005.

BRAGA, R. et al. EFACTAGENT: Agent architecture for management, search and retrievel of e-learning artifacts, Iadis International Conference WWW/Internet 2011. Proceedings of the WWW/Internet 2011. p.468-472. Rio de Janeiro, Brazil.

BRESCIANI, P. et al. TROPOS: An agent-oriented software development methodology, Autonomous Agents and Multi-Agent Systems, v. 8, n. 3, 2004. p. 203-236, Trento, Itália.

BRUSILOVSKY, P.; PEYLO, C. Adaptive and intelligent Web-based educational systems. In: International Journal of Artificial Intelligence in Education, 13, 2-4, Special Issue on Adaptive and Intelligent Web-based Educational Systems, 2003. p. 159-172, Pittsburg, EUA.

BURKE, R. Hybrid recommender systems: survey and experiments. In: User Modeling and User-Adapted Interaction, v. 12, n. 4, Hingham, MA, EUA, 2002. p.331-370.

CAMPOS, F. C. A. et al. Expanding Access to Distance Learning using Learning Objects In: Education in a technological world: communicating current and emerging research and technological efforts. ed. Badajoz : : FORMATEX RESEARCH CENTER, v.1.

_____. Projeto BROAD: Busca semântica por objetos de aprendizagem. In: ESUD 2011 – VIII Congresso Brasileiro de Ensino Superior a Distância Ouro Preto - UNIREDE, Ouro Preto, Brasil, 2011. p. 3–5.

_____. BROAD Project: Semantic Search and Application of Learning Objects. IEEE Technology and Engineering Education (ITEE). Dezembro. 2012

_____. Rede de Ontologias: apoio semântico a linha de produtos de objetos de aprendizagem. In: Simpósio Brasileiro de Informática na Educação - SBIE, Anais do XXIII Simpósio Brasileiro de Informática na Educação, Rio de Janeiro, Brasil, 2012. p. 3–7.

CAZELLA, S. C.; NUNES, M. A. S. N.; REATEGUI, E. A Ciência da Opinião: Estado da Arte em Sistemas de Recomendação. In: André Ponce de Leon F. de Carvalho; Tomasz Kowaltowski. (Org.). Jornada de Atualização de Informática - JAI 2010- CSBC2010. v. 1, p. 161-216, PucRIO, Rio de Janeiro, Brasil, 2010.

CAZELLA, S. C. et al. Recomendação de Objetos de Aprendizagem Empregando Filtragem Colaborativa e Competências In: Simpósio Brasileiro de Informática na Educação - SBIE, Anais do XX Simpósio Brasileiro de Informática na Educação, Brasil, 2009.

CHEN, W.; MIZOGUCHI, R. Communication Content Ontology for Learner Model Agent in Multi-agent Architecture. In: Proceedings of the 7th International Conference on Computers in Education - ICCE99, Osaka, Japan, 1999. p. 95-102.

CHEN, G.; KOTZ, D. A. Survey of Context-Aware Mobile Computing Research. In: Dartmouth Computer Science Technical Report TR2000-381, New Hampshire, EUA, 2000. p.5–9

DAM, K. H.; WINIKOFF, M. Comparing agent-oriented methodologies, In: Proceedings of the Fifth International Bi-Conference Workshop on Agent-Oriented Information Systems held in Melbourne, Melbourne, Australia, 2003. p. 78-93

DUBLIN CORE. Dublin Core Metadata Initiative. Disponível em: <http://dublincore.org/>. Acesso em: 10 de agosto de 2012. 2001.

DEUS, W. J.; CABAL, E. P. Processos de comunicação entre agentes inteligentes para o sistema AgenTCC. In: Simpósio Brasileiro de Informática na Educação - SBIE, Anais do XXIII Simpósio Brasileiro de Informática na Educação, Fortaleza. Brasil, 2012.

DEY, A. K. Understanding and Using Context, Personal and Ubiquitous Computing, v. 5, n. 1, Atlanta, GA, EUA, 2001. p. 4-7.

DEY, A. K.; ABOWD, G. D. Towards a better understanding of context and context-awareness. Gellersen, H.W., ed. 1st International Symposium on Handheld and Ubiquitous Computing (HUC 99). Lecture Notes in Computer Science, 1707. Springer, Verlag, Berlin, Germany, 1999. p. 304-307.

DOLOG, P.; NEJDL, W. Semantic Web Technologies for the Adaptive Web. The Adaptive Web, Springer-Verlag, Heidelberg, Berlin, Germany, 2007. p. 697-719.

DUTRA, R. L. S.; TAROUCO, L. M. R. Objetos de a-prendizagem: uma comparação entre SCORM e IMS learning design. In: Centro Interdisciplinar de Novas Tecnologias na Educação, Universidade Federal do Rio Grande do Sul, Brasil, 2006.

DWIVEDI, P.; BHARADWAJ. K. K. E-learning recommender system for learners in online social networks through association retrieval, In: Proceedings of the CUBE International Information Technology Conference, 6, Maharashtra, India, 2012. p. 676-681.

EDUPERSON. EduPerson Object Class Specification (200210). Disponível em: < http://middleware.internet2.edu/eduperson>. 2005, Acesso em: 29 de agosto 2013.

ELAMY, A. H. Perspectives in agent-based technology, In: AgentLink News, n. 18, 2005. p. 19-22.

ESCOBAR, E; LEMKE, A. P; BLOIS, M. R. Semanticore: Permitindo o desenvolvimento de aplicações baseadas em agentes na web semântica, In: Second Workshop on Software Engineering for Agent-oriented Systems (SEAS), Florianópolis, SC, Brasil, 2006.

EVANS, R. et al. MESSAGE: Methodology for Engineering Systems of Software Agentes, In: Relatório Técnico, European Institute for Research and Strategic Studies in Telecommunications, EURESCOM, EDIN, 2001. p. 0223-0907.

FERRO, M. R. C. et al. Um Modelo de Sistema de Recomendação de Materiais Didáticos para Ambientes Virtuais de Aprendizagem In: Simpósio Brasileiro de Informática na Educação - SBIE, Anais do XXII Simpósio Brasileiro de Informática na Educação, Aracaju, Brasil, 2011.

FININ, T. et al. Specification of the KQML agent-communication language, In: G Proceedings of the Third International Conference on Information and Knowledge Management - CIKM, Gaithersburg, Maryland, EUA, 1994.

FILIPPETTO, A. S. et al. Metodologia INGENIAS para o Desenvolvimento de Sistemas Multi-Agentes, In: Anais da SemInfo, Seminário de Informática, São Leopoldo, RS, Brasil, 2007

FIPA. Foundation for intelligent physical agents (FIPA): Agent management specification, Geneva, Switzerland, 2004.

FLEISCHMANN, A. M. P. Sensibilidade à Situação em Sistemas Educacionais na Web, Tese de Doutorado, Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, RS, Brasil, 2012.

FONER, L. N. What's an Agent Anyway? In: A Sociological Case Study. MIT Media Laboratory, Cambridge, MA, EUA, 1993.

FRITZEN, E.; SIQUEIRA, S. W. M.; ANDRADE, L. C. V. Recuperação Contextual de Informação na Web para Apoiar Aprendizagem Colaborativa em Redes Sociais. In: Simpósio Brasileiro de Informática na Educação, 2012, Anais do Simpósio Brasileiro de Informática na Educação, Rio de Janeiro, Brasil, 2012.

GARCÍA, F. S. Sistema Basado en Tecnologías del Conocimento para Entornos de Servicios Web Semánticos, Tese de Doutorado, Departamento de Ingenier ía de la Información y las Comunicaciones, Universidad de Murcia, España, 2007.

GARCÍA, E. et al. An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering. In: User Modeling and User-Adapted Interaction, 19(1-2), Córdoba, España, 2009. p. 99–132.

GASPARI, M.; MOTTA, E. Symbol-level requirements for agent-level programming, In: ECAI94 the 11th European Conference on Artificial Intelligence, John Wiley and Sons Ltd, Bologna, Italy, 1994. p. 364-368.

GASPARINI, I. et al. Modelo do usuário sensível ao contexto cultural em um sistema e-learning adaptativo. Informática na Educação In: Informática na educação: teoria & prática, v. 14, Brasil, 2011. p. 123-135.

GLUZ, J. C.; VICARI, R. M. MILOS: Infraestrutura de Agentes para Suporte a Objetos de Aprendizagem OBAA, In: Anais do XXI Simpósio Brasileiro de Informática na Educação, SBIE, João Pessoa, PB, Brasil, 2010. p. 2–9.

GÔMES-SANZ, J; J, FUENTES, R. Agent Oriented Software Engineering with INGENIAS, Agented-Oriented Methodologies, In: IDEA Group Publishing, Madrid, España, 2002.

GOSLING, J. The Java Language Specification, 3° edic. In: Addison-Wesley Professional, 2011.

GRUBER, T. A. Translation Approach to Portable Ontology Specifications. 1993. Disponível em: <http://tomgruber.org/writing/ontolingua-kaj-1993.pdf.> Acesso em: 01 de setembro 2013.

GRUPO GRASIA. Disponível em: http://grasia.fdi.ucm.es/main Acesso em: 01 agosto 2013.

HINZ, V. T.; GASPARINI, I.; PIMENTA, M. S. Em busca de recomendação de objetos de aprendizagem em um ambiente e-learning com uso de ontologias - In: LACLO 2011 - Sexto Congresso Latinoamericano de Objetos de Aprendizagem, LACLO 2011 - Sexto Congresso Latinoamericano de Objetos de Aprendizagem, Montevidéu, Uruguai, 2011.

HODGE, G. Metadata made simpler: a guide for libraries, Bethesda, In: MD: National Information Standards Organizations, Bethesda, USA, 2001. 15p.

HODGINS, H. W. The future of learning objects. In: IEEE Learning Technology Standards Committee – LTSC - Learning Object Metadata, Stanford, California, EUA, 2001.

HODGINS, W.; CONNER, M. Everything you ever wanted to know about learning standards but were afraid to ask. Disponível em: <http://www.learnativity.com/standards.html>. Acesso em: 2 de setembro de 2012. Stanford, California, USA, 2001.

HODGINS, W.; DUVAL, E. Learning Technology Standards Committee (LTSC), In: Draft Standard of Learning Object Metadata, New York, NY, EUA, 2002.

IEEE. LOM - Learning Object Metadata Working Group 12. Disponível em: http://www.ieeeltsc.org:8080/Plone/working-group/ Acesso em: 7 de setembro de 2012.

learning-object-metadata-working-group-12/learning-object-metadata-lom-working-group-12, 2012.

IMS. Global Learning Consortium, Inc., IMS Learning Resource Metadata Information Model Disponível em: <http://www.imsglobal.org/>. Acesso em: 21 de julho de 2013, 2002.

IMS LIP. IMS Learner Information Packaging Information Model Specification. Disponível em: http://www.imsglobal.org/profiles/lipinfo01.html, 2007. Acesso em: 01 de setembro 2013.

JÁCOME JÚNIOR, L.; MENDES NETO, F. M.; SILVA, L. C. N. Uma Abordagem Baseada em Algoritmo Genético para Recomendação de Objetos de Aprendizagem Sensível ao Contexto do Estudante. In: Simpósio Brasileiro de Informática na Educação - SBIE, Anais do XXIII Simpósio Brasileiro de Informática na Educação, Fortaleza, Brasil, 2012. p. 4–8.

JOHNSON, L. F. Elusive Vision: Challenges Impeding the Learning Object Economy. Macromedia White Paper. In: New Media Consortium. Disponível em: http://www.nmc.org/pdf/Elusive_Vision.pdf, San Franscisco, CA, EUA, 2003. p. 5–15.

KAY, R.; KNAACK, L. A formative analysis of individual differences in the effectiveness of learning objects in secondary school. In: Computers & Education. 51, 1, Ontario, Canada, 2008. p. 1304–1320.

KLAŠNJA-MILICEVIC, A. et al. Integration of recommendations and adaptive hypermedia into java tutoring system, In: Computer Science and Information Systems, 8(1), 2011. p. 211-224.

LICHTNOW, D. et al. O uso de técnicas de recomendação em um sistema para apoio à aprendizagem colaborativa. Revista Brasileira de informática na educação - RBIE, 14(3), Brasil, 2006. p. 49–59.

LUCENA, P. SemanticAgent, uma plataforma para desenvolvimento de agentes inteligentes, Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - USP, São Paulo, Brasil, 2003.

MAGANO, J. O e-learning no ensino superior: um caso de estudo. Educação, In: Formação Tecnologias; vol. 1(1), Portugal, 2008. p. 79-92. Disponível em http://eft.educom.pt.

MANOUSELIS, N, H. et al. Recommender Systems in Technology Enhanced Learning, Recommender Systems Handbook: A Complete Guide for Research Scientists and Practitioners, L. Rokach, B. Shapira, P. Kantor, and F. Ricci, eds., Springer, 2011. p. 387-409.

MARTINS, N. J. H. et al. Sistema de Recomendação Híbrido para Bibliotecas Digitais que Suportam o Protocolo OAI-PMH In: Simpósio Brasileiro de Informática na Educação - SBIE, Anais do XXII Simpósio Brasileiro de Informática na Educação, Aracaju, Brasil, 2011.

MENDES, L. F. C. Uma arquitetura orientada a agentes para gerenciamento, busca e recuperação de artefatos cientícos, Dissertação de Mestrado, Faculdade de Engenharia Civil, Universidade Federal de Juiz de Fora, Juiz de Fora, Brasil, 2009.

MOORE, P. et al. Intelligent context for personalized m-learning, In International Conference on Complex, In: Intelligent and Software Intensive Systems, 2009.CISIS '09, 2009. p. 247–254.

MOORE, P.; HU, B.; WAN, J. A Context Ontology for Pervasive Mobile Computing, In: The Computer Journal, Vol. 53, 2008. p. 191-207.

MORALES, R.; AGÜERA, A. S. Dynamic sequencing of learning objects. In: Instituto de Investigaciones Eléctricas, México, 2012.

NERY, T. et al. BROAD Project: Semantic Search and Application of Learning Objects, In: IEEE Technology and Engineering Education (ITEE), vol.7, n.3, 2011. p. 3–5.

NIEDERÉE, A. et al. Ontologically Enriched User Profiling for Cross System Personalization. In: User Modelling, Edinburgh. Proceedings [S.l.]: ACM Press. 2005.

NUNES, M. A. S. N. Recommender Systems based on Personality Traits: Could human psychological aspects influence the computer decision-making process. In: 1. ed. Berlin: VDM Verlag Dr. Müller. v.1, Dublin, Ireland, 2009. p. 140.

NWANA, H. S. Software agents: an overview, In: Knowledge Engineering Review, v. 11, n. 3, 1996. p. 1-40.

OBAA. Projeto de objetos de aprendizagem baseado em agentes. Disponível em: http://www.portalobaa.org/. Acessado em: 18 de agosto de 2012, 2010.

ODELL, J. P. H. V. D.; BAUER, B. Extending UML for agents, In: Agent-Oriented Information Systems, In: Workshop at the 17th National conference on Artificial Intelligence, Austin, TX, USA, 2000. p. 3-17.

PADGHAM, L.; WINIKOFF, M. Prometheus: A methodology for developing intelligent agents, In: Agent-Oriented Software Engineering III, vol. 2585 de Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Germany, 2003. p. 174-185.

PAPI. IEEE P1484.2/D7. Draft standard for learning technology. Public and Private information para learners. Disponível em: <*ltsc.ieee.org/wg1/files/ltsa_06.doc* />. Acesso em: Acesso em: 30 de agosto 2013.

PAVÓN, J. et al. The INGENIAS Methodology and Tools, In: cap. 9, Agented-Oriented Methodologies, IDEA Group Publishing, Hershey, USA, 2005. p. 239-276.

POKAHR, A.; BRAUBACH, L. Jadex user guide, In: Technical Report 0.96, University of HAMBURG, HAMBURG, GERMANY, 2007.

REZENDE, P. et al. BROAD-RS: uma arquitetura para recomendação de objetos de aprendizagem sensível ao contexto usando agentes, In: Anais do ESUD 2013 – X Congresso Brasileiro de Ensino Superior a Distância – UNIREDE, Belém/PA, Brasil, 2013.

RODOLPHO, E. R. Convergência digital de objetos de aprendizagem SCORM, Dissertação de Mestrado, In: Universidade Estadual Paulista Júlio de Mesquita Filho -UNESP, São José do Rio Preto, SP, Brasil, 2009.

RUSSEL, S.; NORVIG, P. Artificial Intelligence: A Modern Approach, cap. 2, In: Prentice-Hall Inc, New Jersey, EUA, 1995. p. 31-52.

SABA. UTF: Universal Learner Format – Technical Specification Version 1.0 - 2000 Disponível em: <http://xml.coverpages.org/ulfSpecification20001204.pdf>. Acesso em: 29 de agosto 2013.

SANTOS, D. R. A comparative study of multi-agent systems development methodologies, In: Second Workshop on Software Engineering for Agent-oriented Systems - SEAS, Florianópolis, SC, Brasil, 2006.

SANTOS, N. S. R. S. et al. Metadados para Objetos de Aprendizagem: prova de conceito do modelo UMBRELO. In: Simpósio Brasileiro de Informática na Educação - SBIE, Anais do XIX Simpósio Brasileiro de Informática na Educação, Fortaleza. Brasil, 2012.

_____. Uma análise da estrutura de informação de Objetos de Aprendizagem. In: Simpósio Brasileiro de Informática na Educação - SBIE, Anais do XIX Simpósio Brasileiro de Informática na Educação, Fortaleza, Brasil, 2008a.

SANTOS, N. et al. Digital Libraries and Ontology. In: Yin-Leng Theng; Schubert Foo; Dion Hoe-Lian Goh; Jin-Cheon Na. (Org.). Handbook of Research on Digital Libraries: Design, Development, and Impact. Hershey, Idea Group Inc, v. 1, Pensilvania, EUA, 2008b, p. 206-214.

SCHAFER, J. B. et al. Recommender systems in e-commerce. In: ACM Conference on Eletronic Commerce, Proceeding Denver: ACM. Denver, Colorado, EUA,p. 158-166, 1999.

SCORM. SCORM: Sharable Content Object Reverence Model. 2004. Disponível em: < http://scorm.com>. Acesso em: 15 de agosto de 2012.

SOUZA, A. C., et al. Metadados Educacionais para Objetos de Aprendizagem: Projeto BROAD. In: Congreso Iberoamericano de Informática Educativa, 2010, Santiago. Anais do RIBIE 2010, v. 1. p. 653-659.

SUDHANA, K. M. An Ontology-based Framework for Context-aware Adaptive E-learning System. In: International Conference on Computer Communication and Informatics (ICCCI -2013), Jan. 04–06, 2013, Coimbatore, Índia.

TAROUCO, L. M. R.; FABRE, M. J. M.; TAMUSIUNAS, F. R., 2003. Reusabilidade de objetos educacionais. In: http://www.cinted.ufrgs.br/, In: Centro Interdisciplinar de Novas Tecnologias na Educação, Universidade Federal do Rio Grande do Sul, Brasil, 2003. p. 3–4.

TAVARES, J. C. S. Um modelo para avaliação de aprendizagem no uso de ferramentas síncronas em ensino mediado pela Web, Tese de Doutorado, In: Programa de Pós-Graduação em Informática da PUC-Rio, Rio de Janeiro, Brasil, 2004.

TAYLOR, C. An introduction to metadata. university of queesland library. Disponível em:<http://www.itb.hu/fejlesztesek/meta/hgls/core/Background/An_Introduction_to_Metadata.htm>. Acesso em: 17 de dezembro de 2012, 2003.

TUGSGEREL, B.; ANANE, R.; THEODOROPOULOS, G. An integrated approach to learning object sequencing. In: 10th IEEE International Conference on Advanced Learning Technologies, Birmingham, UK, 2010.

VAN AART, C. Organizational Principles for Multi-Agent Architectures, Series: WSSAT – Whitestein Series in Software Agent Technologies, Birkhauser Verlag, Basel - Boston - Berlin; 2005, 216 pages. ISBN 3-7643-7213-2, 2005.

VCARD. VCARD: The Electronic Business Card Version 2.1 Specification, 1996. Disponível em: <http://www.imc.org/pdi/>. Acesso em: 29 de agosto 2013.

VERBERT, K. et al. Context-Aware Recommender Systems for Learning: A Survey and Future Challenges," IEEE Transactions on Learning Technologies, vol. 99, no. PrePrints, 2012.

VICARI, R. M. et al. Brazilian proposal for agent-based learning objects metadata standard - OBAA. In: http://www.portalobaa.org, Universidade Federal do Rio Grande do Sul, Brasil, 2009. p. 2–5.

VIEIRA, V.; TEDESCO, P.; SALGADO, A. C. Modelos e processos para o desenvolvimento de sistemas sensíveis ao contexto. In: Technical Report, CIn – UFPE, Pernanbuco, Brasil, 2009.

WANG, S, WU C. Application of context-aware and personalized recommendation to implement an adaptive ubiquitous learning system, Expert Systems with Applications 38, 2011. p. 10831–10838.

WILEY, D. A. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. Utah State University, Digital Learning Environments Research Group, Utah, EUA, 2001.

WOOD, M.; DELOACH, S. A. An overview of the multiagent systems engineering methodology, In: The First International Workshop on Agent-Oriented software Engineering - AOSE, Berlin, Germany, 2000. p. 207-222.

WOOLDRIDGE, M. An Introduction to MultiAgent Systems - Second Edition, In: 2º edic., John Wiley and Sons, Liverpool, UK, 2009, ISBN 0470519460.

_____. Intelligent Agents, Boston, UK, 1999. p. 27–77.

WOOLDRIDGE, M., JENNINGS, N. R.; KINNY, D. The gaia methodology for agent-oriented analysis and design, In: Journal of Autonomous Agents and Multi-Agent Systems, v. 3, Boston, UK, 2000. p. 285-312.

YU, E. S., 2011. A Developers Guide to the Semantic Web. In: Springer, Atlanta, EUA, 2011, cap 2, 4 e 5.

ZAÍNE, O. R.. Building a recommender agent for e-learning systems, In: Computers in Education, Proceedings. International Conference on, vol.1 Alberta, Canada, 2002. p. 55-59.

ZANETTE, E. et al. A produção do material didático no contexto cooperativo e colaborativo

da disciplina de cálculo diferencial e integral I, na modalidade de educação a distância na graduação, RENOTE 4.1, 2006.

ZHUHADAR, L. et al. Multi-model ontology based hybrid recommender system in e-learning domain. In: Web Intelligence and Intelligent Agent Technologies, WI-IAT ’09. IEEE/WIC/ACM International Joint Conferences on, volume 3, EUA, 2009. p. 91–95.